VOTRE spécial ORDINATEUR

LE MAGAZINE DE L'INFORMATIQUE A LA MAISON

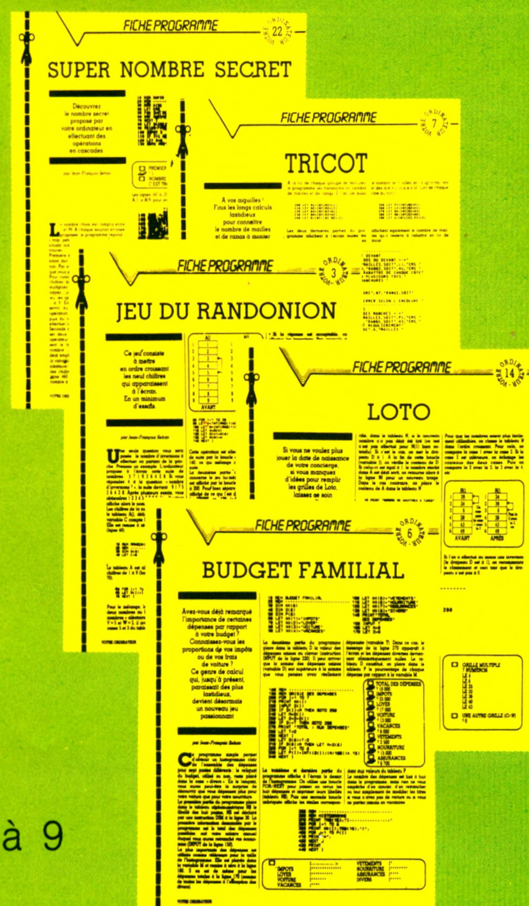
ISSN 0752-2363

HORS SERIE 25

32 FICHES PROGRAMMES UNIVERSELLES

le BASIC par la PRATIQUE

Recueil des fiches parues dans Votre Ordinateur n°1 à 9



JEU ENSEIGNEMENT VIE PRATIQUE MUSIQUE DESSIN

Sur Alice - Apple - Atari - Atmos - Oric 1 - Commodore 64 - Vic 20 - Hector II HR
Thomson MO 5 - TO 7 - TO 7-70 - MSX - Yeno SC 3000 - ZX Spectrum et ZX 81.

en vente chez votre marchand de journaux

PAPIER PEINT

Vous désirez retapisser votre salle à manger ? Placez votre ordinateur à côté du seau de colle. Fini les calculs fastidieux, voilà un moyen de réaliser peut-être des économies.

Enfin un programme simple qui donne le nombre de rouleaux et le découpage des lés, le coût global de l'opération et la valeur des pertes. Il comporte trois parties, mais n'accepte ni les fenêtres ni les portes.

La première partie place les cinq données que vous frappez au clavier (prix du rouleau, largeur du papier, longueur des rouleaux, périmètre de la pièce, hauteur des murs) dans les variables A à E (voir liste au dos).

PRIX DU ROULEAU
? 23
LARGEUR DU PAPIER
? 0,6
LONGUEUR DES ROULEAUX
? 10
PERIMETRE DE LA PIECE
? 18
HAUTEUR DES MURS
? 2,5

```
10 REM PAPIER PEINT
20 REM -----------------
30 PRINT "PRIX DU ROULEAU "
40 INPUT A
50 PRINT "LARGEUR DU PAPIER "
60 INPUT B
70 PRINT "LONGUEUR DES
   ROULEAUX "
80 INPUT C
90 PRINT "PERIMETRE DE LA
   PIECE "
100 INPUT D
110 PRINT "HAUTEUR DES MURS "
120 INPUT E
```

Ces formalités accomplies, le programme calcule dans sa deuxième partie les variables suivantes :

- **Variable F (nombre de lés par rouleau) :** la longueur d'un rouleau étant contenue dans la variable C, l'ordinateur la divise par la hauteur des murs. Pour obtenir un nombre entier de lés, il emploie la fonction INT qui supprime les décimales.
- **Variable G (longueur de chute par rouleau) :** maintenant qu'il connaît le nombre de lés par rouleau (variable F), il calcule la longueur de la chute en multipliant la hauteur des murs par F et en y retranchant la longueur du rouleau.
- **Variable H (nombre de rouleaux) :** en utilisant le périmètre de la pièce, le nombre de lés par rouleau et la largeur du papier, on obtient le nombre de rouleaux. Comme pour C, on utilise la fonction INT pour éviter d'avoir des fractions (bonjour monsieur, je voudrais 3,75 rouleaux de papier bleu !). La valeur 0.99999 ajoute un rouleau de plus dans le cas où le dernier est incomplet. Exemples : si on a 5 rouleaux complets INT(5+.99999) donne 5 rlx (rouleaux) ; si on a 5.75 rouleaux INT(5.75+.99999) donne 6 rlx (rouleaux).
- **Variable I (nombre total de lés) :** on utilise la même méthode que pour le calcul de G : INT (nombre de lés +.99999).
- **Variable J (nombre de lés sur le dernier rouleau) :** découle du nombre de lés et du nombre de rouleaux.
- **Variable K (longueur de la chute sur le dernier rouleau) :** découle du nombre de lés sur le dernier rouleau et de la longueur d'un rouleau.
- **Variable L (prix total du papier peint) :** obtenu en multipliant le nombre de rouleaux par son prix.
- **Variable M (longueur des chutes) :** donne le total des chutes par rouleau et sur le dernier incomplet.
- **Variable N (pertes en pourcentage) :** total des chutes divisé par la longueur totale du papier et divisé par 100 pour obtenir un pourcentage.

```
130 REM -----------------
140 REM CALCULS
150 LET F=INT(C/E)
160 LET G=C-F*E
170 LET H=INT((D/(F*B))+.99999)
180 LET I=INT((D/B)+.99999)
190 LET J=I-((H-1)*F)
200 LET K=C-(J*E)
210 LET L=H*A
220 LET M=((H-1)*G)+K
230 LET N=M/((H*C)/100)
```

La troisième et dernière partie du programme sert à afficher le résultat de tous les calculs précédents. L'ordinateur chiffre alors la valeur des pertes en fonction du prix global et du pourcentage de chute de papier, puis il affiche à l'écran le résultat final de ses cogitations.

```
240 REM -----------------
250 REM RESULTATS
260 PRINT
270 PRINT "NB DE RLX:";H
280 PRINT "NB DE LES PAR RLX:";F
290 PRINT "IL Y A ";H-1;" CHUTES DE ";G;" METRES"
300 PRINT "PLUS ";K;" METRES SUR LE DERNIER RLX"
310 PRINT "IL RESTE DONC ";M;" METRES DE CHUTES POUR LES RACCORDS"
320 PRINT "LE PRIX TOTAL EST DE ";L;" FRANCS"
330 PRINT "ET LES PERTES DE ";N;" %"
340 PRINT "SOIT ";L*N/100;" FRANCS"
```

NB DE RLX : 8
NB DE LES PAR RLX : 4
IL Y A 7 CHUTES DE 0 METRE
PLUS DE 5 METRES SUR LE DERNIER RLX
IL RESTE DONC 5 METRES DE CHUTES POUR LES RACCORDS
LE PRIX TOTAL EST DE 184 FRANCS
ET LES PERTES DE 6.25 %
SOIT 11.5 FRANCS

FICHE PROGRAMME

PAPIER PEINT - PAPIER PEINT - PAPIER PEINT

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10 REM PAPIER PEINT
20 REM -----------------
30 PRINT "PRIX DU ROULEAU "
40 INPUT A
50 PRINT "LARGEUR DU PAPIER "
60 INPUT B
70 PRINT "LONGUEUR DES ROULEAUX "
80 INPUT C
90 PRINT "PERIMETRE DE LA PIECE "
100 INPUT D
110 PRINT "HAUTEUR DES MURS "
120 INPUT E
130 REM -----------------
140 REM CALCULS
150 LET F=INT(C/E)
160 LET G=C-F*E
170 LET H=INT((D/(F*B))+.99999)
180 LET I=INT((D/B)+.99999)
190 LET J=I-((H-1)*F)
200 LET K=C-(J*E)
210 LET L=H*A
220 LET M=((H-1)*G)+K
230 LET N=M/((H*C)/100)
240 REM -----------------
250 REM RESULTATS
260 PRINT
270 PRINT "NB DE RLX:";H
280 PRINT "NB DE LES PAR RLX:";F
290 PRINT "IL Y A ";H-1;" CHUTES
    DE ";G;" METRES"
300 PRINT "PLUS ";K;" METRES SUR LE
    DERNIER RLX"
310 PRINT "IL RESTE DONC ";M;" METRES DE
    CHUTES POUR LES RACCORDS"
320 PRINT "LE PRIX TOTAL EST DE ";L;" FRANCS"
330 PRINT "ET LES PERTES DE ";N;" %"
340 PRINT "SOIT ";L*N/100;" FRANCS"
```

Liste des variables de A à N

- **A** Prix du rouleau
- **B** Largeur du papier
- **C** Longueur des rouleaux
- **D** Périmètre de la pièce
- **E** Hauteur des murs
- **F** Nombre de lés par rouleau
- **G** Longueur de chute par rouleau
- **H** Nombre de rouleaux
- **I** Nombre total de lés
- **J** Nombre de lés sur le dernier rouleau
- **K** Longueur de la chute sur le dernier rouleau
- **L** Prix total du papier peint
- **M** Longueur des chutes
- **N** Pertes en pourcentage.



CALENDRIER PERPÉTUEL

Pompiers, éboueurs, postiers risquent de faire la grimace en fin d'année. Qu'ils ne s'inquiètent pas : notre programme permet seulement de prévoir ses vacances, ponts ou rendez-vous plus professionnels, mais en aucun cas son biorythme

Entrons dans le vif du sujet : chaque mois comporte un nombre fixe de jours (exception faite du mois de février, qui confirme la règle). Après avoir dimensionné le tableau M [DIM M (12)] situé en ligne 30, remplissez-le des nombres de jours par mois. Pour février, le programme se contente, dans sa première partie, de 28 jours, les années bissextiles étant calculées par la suite. Pour que le calendrier soit complet, on indique le nombre de jours passés et à venir. C'est le rôle de la variable J que l'on met au début à zéro (ligne 40).

```
10 REM CALENDRIER
20 REM -------------
30 DIM M(12)
40 LET J=0
50 LET B=0
60 LET M(1)=31
70 LET M(2)=28
80 LET M(3)=31
90 LET M(4)=30
100 LET M(5)=31
110 LET M(6)=30
120 LET M(7)=31
130 LET M(8)=31
140 LET M(9)=30
150 LET M(10)=31
160 LET M(11)=30
170 LET M(12)=31
```

La seconde partie du programme calcule le jour de la semaine au premier janvier, et demande, comme base de départ, l'année de l'édition [INPUT de la ligne 190] qu'il place dans la variable U dès que l'écran affiche ANNÉE DU CALENDRIER ? tapez au clavier l'année choisie suivie de RUN. Après les calculs des lignes 210 à 240, on trouve dans la variable Y le nombre 1 pour lundi, 2 pour mardi, etc. Pour déterminer les années bissextiles, le programme divise par quatre la variable U. S'il obtient un nombre entier, il s'agit bien d'une année de trois cent soixante-six jours (la ligne 250 effectue l'opération). En cas de résultat positif, il place le nombre 1 dans la variable B

```
175 REM -------------------
176 REM CALCUL JOUR DE LA SEMAINE
180 PRINT"ANNEE DU CALENDRIER ? "
190 INPUT U
200 IF U>99 THEN GOTO 180
210 LET P=(U-1)/100
220 LET Q=INT(P/4)
230 LET M=36+INT((5*(U-1))/4)-INT(P)+Q
240 LET Y=M-(7*INT(M/7))+1
250 IF INT(U/4)<>U/4 THEN GOTO 270
260 LET B=1
```

La troisième et dernière partie du programme édite à l'écran le calendrier mois par mois. Il se réfère à la variable B pour compter 29 jours au lieu de 28 pour le mois de février.
Chaque jour s'affiche sur l'écran à la position auparavant déterminée par la variable Y (instruction TAB de la ligne 350). Celle-ci est incrémentée (on lui ajoute 1) après chaque jour affiché. Dès qu'elle dépasse 8, on la remet à 1 (lundi).
Quand le mois est entièrement affiché, l'ordinateur attend que vous appuyiez sur la touche « retour chariot » (ENTER ou RETURN sur la plupart des ordinateurs domestiques), avant d'afficher le mois suivant (NEXT de la ligne 450) ; le programme incrémente la variable J du nombre de jours passés.

```
265 REM --------------------
266 REM EDITION CALENDRIER
270 FOR I=1 TO 12
280 LET N=M(I)
290 IF I<>2 THEN GOTO 320
300 IF B<>1 THEN GOTO 320
310 LET N=29
320 PRINT TAB(10);J;"-";I;U;"-";365+B-J
330 PRINT"    L   M   M   J   V   S   D"
340 FOR K=1 TO N
350 PRINT TAB(Y*4);K;
360 LET Y=Y+1
370 IF Y<8 THEN GOTO 400
380 PRINT
390 LET Y=1
400 NEXT K
410 PRINT
420 PRINT"APPUYEZ SUR UNE TOUCHE"
430 INPUT R$
440 LET J=J+N
450 NEXT I
```



```
           31 - 2  84 - 335
L    M    M    J    V    S    D
          1    2    3    4    5
6    7    8    9    10   11   12
13   14   15   16   17   18   19
20   21   22   23   24   25   26
27   28   29
```

APPUYEZ SUR UNE TOUCHE

CALENDRIER PERPÉTUEL-CALENDRIE

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10 REM CALENDRIER
20 REM -------------
30 DIM M(12)
40 LET J=0
50 LET B=0
60 LET M(1)=31
70 LET M(2)=28
80 LET M(3)=31
90 LET M(4)=30
100 LET M(5)=31
110 LET M(6)=30
120 LET M(7)=31
130 LET M(8)=31
140 LET M(9)=30
150 LET M(10)=31
160 LET M(11)=30
170 LET M(12)=31
175 REM -------------------
176 REM CALCUL JOUR DE LA SEMAINE
180 PRINT"ANNEE DU CALENDRIER ? "
190 INPUT U
200 IF U>99 THEN GOTO 180
210 LET P=(U-1)/100
220 LET Q=INT(P/4)
230 LET M=36+INT((5*(U-1))/4)-INT(P)+Q
240 LET Y=M-(7*INT(M/7))+1
250 IF INT(U/4)<>U/4 THEN GOTO 270
260 LET B=1
265 REM --------------------
266 REM EDITION CALENDRIER
270 FOR I=1 TO 12
280 LET N=M(I)
290 IF I<>2 THEN GOTO 320
300 IF B<>1 THEN GOTO 320
310 LET N=29
320 PRINT TAB(10);J;"-";I;U;"-";365+B-J
330 PRINT"    L   M   M   J   V   S   D"
340 FOR K=1 TO N
350 PRINT TAB(Y*4);K;
360 LET Y=Y+1
370 IF Y<8 THEN GOTO 400
380 PRINT
390 LET Y=1
400 NEXT K
410 PRINT
420 PRINT"APPUYEZ SUR UNE TOUCHE"
430 INPUT R$
440 LET J=J+N
450 NEXT I
```

Liste des variables

- **B** B=1 si l'année est bissextile
- **I** indice de boucle dans l'édition des douze mois de l'année
- **J** nombre de jours écoulés
- **K** indice de boucle pour l'édition des jours
- **M()** tableau du nombre de jours par mois
- **M** utilisée pour le calcul du jour de la semaine
- **N** nombre de jours par mois
- **P** utilisée pour le calcul du jour de la semaine
- **Q** utilisée pour le calcul du jour de la semaine
- **U** année du calendrier
- **Y** numéro du jour de la semaine : lundi=1, mardi=2, etc.



JEU DU RANDONION

Ce jeu consiste à mettre en ordre croissant les neuf chiffres qui apparaissent à l'écran. En un minimum d'essais.

par Jean-François Sehan

Une seule question vous sera posée : le nombre d'inversions à effectuer en partant de la gauche. Prenons un exemple. L'ordinateur propose à l'écran cette suite de nombres : 5 7 1 9 3 6 4 2 8. Si vous répondez 4 à la question « nombre d'inversions ? », la suite devient : 9 1 7 5 3 6 4 2 8. Après plusieurs essais, vous obtiendrez 1 2 3 4 5 6 7 8 9. L'ordinateur affiche alors le nombre de coups.

Les chiffres de la suite sont placés dans le tableau A(), défini à la ligne 30. La variable C compte le nombre de coups. Elle est remise à zéro au départ du jeu (ligne 40).

```
10 REM RANDONION
20 REM ------------------
30 DIM A(9)
40 LET C=0
```

Le tableau A est alors rempli des neuf chiffres de 1 à 9 (boucle des lignes 50 à 70).

```
50 FOR I=1 TO 9
60 LET A(I)=I
70 NEXT I
```

Pour le mélange, le programme utilise deux nombres au hasard V et W (dits nombres « aléatoires »). Par exemple, si V=5 et W=3, il inverse le contenu des cases 5 et 3 du tableau A() :

	A() AVANT		A() APRES
1	1	1	1
2	2	2	2
3	3	3	5
4	4	4	4
5	5	5	3
6	6	6	6
7	7	7	7
8	8	8	8
9	9	9	9

```
80 FOR I=1 TO 20
90 LET V=INT(RND(1)*9)+1
100 LET W=INT(RND(1)*9)+1
110 LET N=A(V)
120 LET A(V)=A(W)
130 LET A(W)=N
140 NEXT I
```

Cette opération est effectuée vingt fois de suite par la boucle des lignes 80 à 140, ce qui mélange correctement la suite.

La deuxième partie du programme concerne le jeu lui-même. Le tableau est affiché par la boucle des lignes 180 à 200. Pour bien séparer ce qui va être affiché de ce qui l'est déjà sur l'écran, le PRINT en 170 insère une ligne blanche. En effet, si un PRINT n'est suivi d'aucun message ou variable, le Basic saute une ligne.

```
150 REM -----------------
160 REM LE JEU
170 PRINT
180 FOR I=1 TO 9
190 PRINT A(I);
200 NEXT I
```

Après le message « nombre d'inversions ? », la ligne 230 place dans la variable R le nombre que vous tapez au clavier. Deux solutions se présentent alors.

• **Si le nombre est inférieur à 2 ou supérieur à 9**, on refuse cette réponse et on saisit à nouveau un autre nombre d'inversions. C'est le rôle des deux lignes de test en 240 et 250.

```
210 PRINT
220 PRINT"NOMBRE D'INVERSIONS ? "
230 INPUT R
240 IF R<2 THEN GOTO 220
250 IF R>9 THEN GOTO 220
```

• **Si la réponse est acceptable**, on effectue les inversions. Par exemple si R contient 6 :

	AVANT		APRES
1	5	1	9
2	7	2	6
3	1	3	3
4	3	4	1
5	6	5	7
6	9	6	5
7	4	7	4
8	2	8	2
9	8	9	8

Les schémas ci-dessus montrent que trois inversions seulement sont effectuées. La boucle des lignes 260 à 300 est donc répétée pour la variable I allant de 1 à R divisée par 2. Pour les inversions, on utilise la même méthode que pour le mélange.

```
260 FOR I=1 TO R/2
270 LET N=A(I)
280 LET A(I)=A(R-I+1)
290 LET A(R-I+1)=N
300 NEXT I
```

Une inversion comptant pour un essai, on incrémente (ajoute plus 1) le compteur C (ligne 310).

```
310 LET C=C+1
```

Le jeu est fini si le tableau est en ordre croissant. Les lignes 320 à 340 testent cette éventualité. Si au moins une case n'est pas dans l'ordre (A(I)=I), on retourne en 170 pour une nouvelle proposition après le test en 330.

```
320 FOR I=1 TO 9
330 IF A(I)<>I THEN GOTO 170
340 NEXT I
```

```
4 5 3 2 1 6 7 8 9
NOMBRE D'INVERSIONS ?
? 2
5 4 3 2 1 6 7 8 9
NOMBRE D'INVERSIONS ?
? 5
VOUS AVEZ TROUVE EN
8 COUPS
```

suite page 78

JEU DU RANDONION - JEU DU R

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

Si c'est gagné, on affiche le nombre de coups (variable C).

```
350 PRINT
360 PRINT "VOUS AVEZ
    TROUVE EN ";C;" COUPS"
370 END
```

Le jeu pourrait être plus compliqué en prenant comme liste de départ des chiffres ou les neuf premières lettres de l'alphabet alternativement. A vos claviers !
Laissez libre cours à votre imagination pour que fleurissent cent mille petits programmes...

```
10 REM RANDONION
20 REM ------------------
30 DIM A(9)
40 LET C=0
50 FOR I=1 TO 9
60 LET A(I)=I
70 NEXT I
80 FOR I=1 TO 20
90 LET V=INT(RND(1)*9)+1
100 LET W=INT(RND(1)*9)+1
110 LET N=A(V)
120 LET A(V)=A(W)
130 LET A(W)=N
140 NEXT I
150 REM ----------------
160 REM LE JEU
170 PRINT
180 FOR I=1 TO 9
190 PRINT A(I);
200 NEXT I
210 PRINT
220 PRINT"NOMBRE D'INVERSIONS ? "
230 INPUT R
240 IF R<2 THEN GOTO 220
250 IF R>9 THEN GOTO 220
260 FCR I=1 TO R/2
270 LET N=A(I)
280 LET A(I)=A(R-I+1)
290 LET A(R-I+1)=N
300 NEXT I
310 LET C=C+1
320 FOR I=1 TO 9
330 IF A(I)<>I THEN GOTO 170
340 NEXT I
350 PRINT
360 PRINT "VOUS AVEZ TROUVE EN ";C;" COUPS"
370 END
```

Table des variables

A	Tableau de la suite	R	Réponse du joueur
C	Compteur d'essais	V	Nombre aléatoire du mélange
I	Indice de boucle	W	Deuxième nombre aléatoire du mélange
N	Variable de stockage pour une inversion		



LE CHIEN DE BERGER

Seriez-vous un bon berger ? Guideriez-vous correctement votre chien pour que les moutons rentrent sagement dans la bergerie ?

par Jean-François Sehan

Ce jeu évolue sur un damier de vingt cases sur douze cases dans lesquelles sont placés aléatoirement, au début, une dizaine de moutons et un chien. En modifiant l'emplacement de ce dernier, les moutons se déplacent automatiquement dans le sens opposé. A vous de trouver les meilleurs emplacements pour que tous les moutons regagnent la bergerie (bord droit du damier) en un minimum d'essais.

La première partie du programme place dans les tableaux X et Y (déclarés par les instructions DIM des lignes 40 et 50) les coordonnées des moutons. On utilise aux lignes 80 et 90 des fonctions RND pour obtenir des nombres aléatoires. Sur certains ordinateurs (ZX-81, par exemple), cette instruction s'écrit RND et non RND (1) comme dans le listing. Une vérification dans le manuel d'utilisation de votre ordinateur s'impose parfois. Les coordonnées du chien sont placées à l'indice onze des tableaux. Le tableau à deux dimensions T$, déclaré à la ligne 30, contient la représentation du damier.

```
10 REM CHIEN DE BERGER
20 REM ---------------------
30 DIM T$(20,12)
40 DIM X(11)
50 DIM Y(11)
60 LET N=0
70 FOR I=1 TO 11
80 LET X(I)=INT(RND(1)*20)+1
90 LET Y(I)=INT(RND(1)*12)+1
100 IF T$(X(I),Y(I))<>"" THEN GOTO 80
110 LET T$(X(I),Y(I))="M"
120 IF I=11 THEN LET T$(X(I),Y(I))="C"
130 NEXT I
```

La deuxième partie du programme affiche à l'écran le damier ainsi que les numéros des cases pour déterminer la position du chien. Le programme demande alors les nouvelles coordonnées X et Y du chien (instruction INPUT des lignes 280 et 300). La ligne 310 vérifie que vos réponses ne sont pas farfelues. Si elles le sont, le damier est de nouveau affiché.

```
140 PRINT "    X                   1 1 1 1 1 1 1 1 1 1 2";
150 PRINT "Y   1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0";
160 PRINT
170 FOR I=1 TO 12
180 IF I<10 THEN PRINT " ";
190 PRINT I;
200 FOR J=1 TO 20
210 PRINT T$(J,I);" ";
220 IF T$(J,I)="" THEN PRINT " ";
230 NEXT J
240 IF I<6 OR I>7 THEN PRINT "I";
250 PRINT
260 NEXT I
270 PRINT "COORDONNEE X DU CHIEN"
280 INPUT X
290 PRINT "COORDONNEE Y DU CHIEN"
300 INPUT Y
310 IF X<1 OR X>20 OR Y<1 OR Y>12 THEN GOTO 140
320 IF T$(X,Y)<>"" AND T$(X,Y)<>"C" THEN GOTO 140
330 LET N=N+1
340 LET T$(X(11),Y(11))=""
350 LET X(11)=X
360 LET Y(11)=Y
370 LET T$(X,Y)="C"
```

La troisième et dernière partie du programme déplace les moutons suivant la position du chien (lignes 400 à 660). Dès qu'un mouton entre dans la bergerie, ses coordonnées X et Y prennent les valeurs O et le compteur E (nombre de moutons éliminés) est incrémenté de 1. Quand le compteur marque 10, le programme affiche le message de la ligne 680 (test IF de la ligne 650). Avant chaque essai, le compteur N est incrémenté de 1 pour connaître le nombre de coups nécessaires pour réussir.

```
   X                   1 1 1 1 1 1 1 1 1 1 2
Y  1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
 1                               M         I
 2                                         I
 3                                         I
 4                                       M I
 5                                         I
 6                   M
 7                     M       M
 8     M                                   I
 9                           M C     M     I
10                                         I
11                                         I
12     M                                 M I
```

```
380 FOR I=1 TO 3
390 LET E=0
400 FOR J=1 TO 10
410 IF X(J)<>0 THEN GOTO 440
420 LET E=E+1
430 GOTO 640
440 LET X=X(J)
450 LET Y=Y(J)
460 IF X(J)<X(11) THEN LET X=X-1
470 IF X(J)=20 AND (Y(J)<6 OR Y(J)>7) THEN LET X=X-1
480 IF X<>X(J) THEN GOTO 500
490 IF X(J)>X(11) OR X(J)=1 THEN LET X=X+1
500 IF Y(J)<Y(11) OR Y(J)=12 THEN LET Y=Y-1
510 IF Y<>Y(J) THEN GOTO 530
520 IF Y(J)>Y(11) OR Y(J)=1 THEN LET Y=Y+1
530 IF X>20 AND Y>5 AND Y<8 THEN GOTO 570
540 IF X<1 OR X>20 OR Y<1 OR Y>12 THEN GOTO 640
550 IF T$(X,Y)<>"" THEN GOTO 640
560 GOTO 600
570 LET X=0
580 LET Y=0
590 LET E=E+1
600 LET T$(X(J),Y(J))=""
610 LET T$(X,Y)="M"
620 LET X(J)=X
630 LET Y(J)=Y
640 NEXT J
650 IF E=10 THEN GOTO 680
660 NEXT I
670 GOTO 140
680 PRINT"VOUS AVEZ REUSSI EN ";N;" COUPS"
```

FICHE PROGRAMME

VOTRE ORDINATEUR 4

JEU DU CHIEN DE BERGER JEU

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10 REM CHIEN DE BERGER
20 REM --------------------
30 DIM T$(20,12)
40 DIM X(11)
50 DIM Y(11)
60 LET N=0
70 FOR I=1 TO 11
80 LET X(I)=INT(RND(1)*20)+1
90 LET Y(I)=INT(RND(1)*12)+1
100 IF T$(X(I),Y(I))<>"" THEN GOTO 80
110 LET T$(X(I),Y(I))="M"
120 IF I=11 THEN LET T$(X(I),Y(I))="C"
130 NEXT I
140 PRINT "     X                   1 1 1 1 1 1 1 1 1 1 2";
150 PRINT "Y    1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0";
160 PRINT
170 FOR I=1 TO 12
180 IF I<10 THEN PRINT " ";
190 PRINT I;
200 FOR J=1 TO 20
210 PRINT T$(J,I);" ";
220 IF T$(J,I)="" THEN PRINT " ";
230 NEXT J
240 IF I<6 OR I>7 THEN PRINT "I";
250 PRINT
260 NEXT I
270 PRINT "COORDONNEE X DU CHIEN"
280 INPUT X
290 PRINT "COORDONNEE Y DU CHIEN"
300 INPUT Y
310 IF X<1 OR X>20 OR Y<1 OR Y>12 THEN GOTO 140
320 IF T$(X,Y)<>"" AND T$(X,Y)<>"C" THEN GOTO 140
330 LET N=N+1
340 LET T$(X(11),Y(11))=""
350 LET X(11)=X
360 LET Y(11)=Y
370 LET T$(X,Y)="C"
380 FOR I=1 TO 3
390 LET E=0
400 FOR J=1 TO 10
410 IF X(J)<>0 THEN GOTO 440
420 LET E=E+1
430 GOTO 640
440 LET X=X(J)
450 LET Y=Y(J)
460 IF X(J)<X(11) THEN LET X=X-1
470 IF X(J)=20 AND (Y(J)<6 OR Y(J)>7) THEN LET X=X-1
480 IF X<>X(J) THEN GOTO 500
490 IF X(J)>X(11) OR X(J)=1 THEN LET X=X+1
500 IF Y(J)<Y(11) OR Y(J)=12 THEN LET Y=Y-1
510 IF Y<>Y(J) THEN GOTO 530
520 IF Y(J)>Y(11) OR Y(J)=1 THEN LET Y=Y+1
530 IF X>20 AND Y>5 AND Y<8 THEN GOTO 570
540 IF X<1 OR X>20 OR Y<1 OR Y>12 THEN GOTO 640
550 IF T$(X,Y)<>"" THEN GOTO 640
560 GOTO 600
570 LET X=0
580 LET Y=0
590 LET E=E+1
600 LET T$(X(J),Y(J))=""
610 LET T$(X,Y)="M"
620 LET X(J)=X
630 LET Y(J)=Y
640 NEXT J
650 IF E=10 THEN GOTO 680
660 NEXT I
670 GOTO 140
680 PRINT"VOUS AVEZ REUSSI EN ";N;" COUPS"
```

Liste des variables

E	Nombre de moutons éliminés	**X**	Nouvelle position en x du chien
I	Indice de boucle	**X()**	Position en x des moutons
J	Indice de boucle	**Y**	Nouvelle position en y du chien
N	Nombre d'essais	**Y()**	Position en y des moutons
T$()	Tableau représentant le damier		



LE FÉMININ DES MOTS

Vous ne connaissez pas bien le féminin de certains noms d'animaux ? Découvrez-les avec ce programme de traduction qui vous aidera, en outre, à maîtriser l'introduction et l'exploitation de données dans un programme.

par Jean-François Sehan

L'ordinateur affiche à l'écran le nom d'un mâle qu'il tire au hasard dans une liste préétablie. Il suffit de taper au clavier le féminin qui lui correspond. Une erreur ? La machine ne vous mettra pas une mauvaise note mais donnera simplement la réponse exacte. Côté programme, c'est très simple. Tous les noms sont placés dans un tableau à deux dimensions. La première colonne donne les noms masculins, la deuxième leurs correspondants féminins.

```
10 REM LES FEMININS
20 REM ----------------
30 DIM M$(10,2)
40 LET M$(1,1)="BELIER"
50 LET M$(1,2)="BREBIS"
60 LET M$(2,1)="CERF"
70 LET M$(2,2)="BICHE"
80 LET M$(3,1)="LEVRIER"
90 LET M$(3,2)="LEVRETTE"
100 LET M$(4,1)="SANGLIER"
110 LET M$(4,2)="LAIE"
120 LET M$(5,1)="VERRAT"
130 LET M$(5,2)="TRUIE"
140 LET M$(6,1)="BOUC"
150 LET M$(6,2)="CHEVRE"
160 LET M$(7,1)="CHEVREUIL"
170 LET M$(7,2)="CHEVRETTE"
180 LET M$(8,1)="ETALON"
190 LET M$(8,2)="JUMENT"
200 LET M$(9,1)="LIEVRE"
210 LET M$(9,2)="HASE"
220 LET M$(10,1)="SINGE"
230 LET M$(10,2)="GUENON"
```

Si l'ordinateur possède les instructions DATA et READ (liste de données, lecture des données), il est possible de modifier le programme ainsi :

```
40 DATA BELIER,BREBIS,
        CERF,BICHE,LEVRIER,
        LEVRETTE,SANGLIER
50 DATA LAIE,VERRAT,TRUIE,
        BOUC,CHEVRE,
        CHEVREUIL,CHEVRETTE
60 DATA ETALON,JUMENT,
        LIEVRE,HASE,
        SINGE,GUENON
70 FOR I=1 TO 10
80 READ M$(I,1),M$(I,2)
90 NEXT I
```

(On supprimera alors les lignes 100 à 230.)
Avec un ZX-81, il faut définir la longueur des chaînes de caractères du tableau M$ en écrivant :
30 DIM M$ (10, 2,9)
La donnée la plus longue comprenant neuf caractères (mot « CHEVREUIL »).
La variable P donne un chiffre au hasard compris entre 1 et 10. La fonction RND (1) donne un nombre toujours compris entre 0 et 0.9999... En multipliant ce dernier par 10 et en ne prenant que la partie entière (c'est le rôle de la fonction INT), on obtient un nombre entre 0 et 9. Il suffit de lui ajouter 1, et le tour est joué. Pour le ZX-81, la fonction RND (1) doit être remplacée par RND.

```
240 REM ----------------
250 REM QUESTIONS
260 LET P=INT(RND(1)*10)+1
270 PRINT
```

Maintenant que ce chiffre « aléatoire » est en votre possession (dans la variable P, bien évidemment !), on affiche le mot du tableau M$ qui y correspond. Prenons un exemple. Si P=6, on affiche M$(6,1) c'est-à-dire « BOUC », le féminin à trouver étant « CHEVRE » (dans M$(6,2).
La fonction INPUT de la ligne 300 saisit votre réponse dans la variable R$. Il suffit de comparer R$ et M$(P,2) pour savoir s'il s'agit d'une réponse correcte. Le programme se sépare alors en deux parties (fonction IF de la ligne 310). Quelle que soit l'issue de votre réponse, on retourne à la ligne 250 pour une nouvelle proposition.

```
280 PRINT"LE MALE: "
    ;M$(P,1)
290 PRINT"LA FEMELLE ? "
300 INPUT R$
310 IF R$=M$(P,2)
    THEN GOTO 340
320 PRINT"NON. C'EST
    LA ";M$(P,2)
330 GOTO 250
340 PRINT"OUI. C'EST
    LA ";M$(P,2)
350 GOTO 250
```

Ce programme n'est pas limité à la comparaison des mâles et des femelles. Il peut être employé dans bien d'autres cas.
Voici une autre possibilité. Modifiez en conséquence les lignes 40 à 230 en utilisant les mots de ce nouveau tableau. N'oubliez pas aussi les libellés des PRINT des lignes 280 et 290.

M$(,)

	1	2
1	*CABIAI*	*MAMMIFERE*
2	*CALAO*	*OISEAU*
3	*CAROLIN*	*OISEAU*
4	*CARABE*	*INSECTE*
5	*CARNELET*	*POISSON*
6	*CASOAR*	*OISEAU*
7	*CHABOT*	*POISSON*
8	*MACHAON*	*INSECTE*
9	*CHEVECHE*	*OISEAU*
10	*CICINDELE*	*INSECTE*

Le tableau M$ est ici limité à dix éléments. Il est possible d'augmenter ce nombre, dans les limites de l'ordinateur. La liste des mots peut ainsi s'allonger jusqu'à 30 ou 40 mots, voire 100 si possible.
A vous maintenant de créer de nouveaux tableaux !

```
LE MALE : CERF
LA FEMELLE ?
? BICHE
OUI. C'EST LA BICHE
LE MALE : CHEVREUIL
LA FEMELLE ?
? BICHE
NON. C'EST LA CHEVRETTE
LE MALE : LIEVRE
LA FEMELLE ?
?
```

LE FÉMININ DES MOTS-LE FÉM

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10 REM LES FEMININS
20 REM ----------------
30 DIM M$(10,2)
40 LET M$(1,1)="BELIER"
50 LET M$(1,2)="BREBIS"
60 LET M$(2,1)="CERF"
70 LET M$(2,2)="BICHE"
80 LET M$(3,1)="LEVRIER"
90 LET M$(3,2)="LEVRETTE"
100 LET M$(4,1)="SANGLIER"
110 LET M$(4,2)="LAIE"
120 LET M$(5,1)="VERRAT"
130 LET M$(5,2)="TRUIE"
140 LET M$(6,1)="BOUC"
150 LET M$(6,2)="CHEVRE"
160 LET M$(7,1)="CHEVREUIL"
170 LET M$(7,2)="CHEVRETTE"
180 LET M$(8,1)="ETALON"
190 LET M$(8,2)="JUMENT"
200 LET M$(9,1)="LIEVRE"
210 LET M$(9,2)="HASE"
220 LET M$(10,1)="SINGE"
230 LET M$(10,2)="GUENON"
240 REM ----------------
250 REM QUESTIONS
260 LET P=INT(RND(1)*10)+1
270 PRINT
280 PRINT"LE MALE: ";M$(P,1)
290 PRINT"LA FEMELLE ? "
300 INPUT R$
310 IF R$=M$(P,2) THEN GOTO 340
320 PRINT"NON. C'EST LA ";M$(P,2)
330 GOTO 250
340 PRINT"OUI. C'EST LA ";M$(P,2)
350 GOTO 250
```

Liste des variables

M$(,) Tableau des questions et des réponses

P Position aléatoire dans le tableau M$(,)

R$ Réponse du joueur



BUDGET FAMILIAL

Avez-vous déjà remarqué l'importance de certaines dépenses par rapport à votre budget ? Connaissez-vous les proportions de vos impôts ou de vos frais de voiture ? Ce genre de calcul qui, jusqu'à présent, paraissait des plus fastidieux, devient désormais un nouveau jeu passionnant

par Jean-François Sehan

Ce programme simple permet d'obtenir un histogramme clair de l'ensemble des dépenses pour sept postes différents ; le reliquat du budget, utilisé ou non, reste placé dans la case « divers ». En le lançant, vous aurez peut-être la surprise de découvrir que vous dépensez plus pour votre voiture que pour votre nourriture.

La première partie du programme place dans le tableau alphanumérique N$ le libellé des huit postes. N$ est déclaré par une instruction DIM à la ligne 30. La première information demandée par le programme est le total des dépenses possibles, soit votre salaire annuel duquel vous aurez retranché vos économies (INPUT de la ligne 150).

La plus importante des dépenses est utilisée comme référence pour la taille de l'histogramme. Elle est placée dans la variable M et remise à zéro à la ligne 160. Il en est de même pour les dépenses totales à la ligne 170 (somme de toutes les dépenses à l'exception des divers).

```
10 REM BUDGET FAMILIAL
20 REM -------------------------
30 DIM N$(8)
40 DIM D(8)
50 DIM P(8)
60 LET N$(1)="IMPOTS"
70 LET N$(2)="LOYER"
80 LET N$(3)="VOITURE"
90 LET N$(4)="VACANCES"
100 LET N$(5)="VETEMENTS"
110 LET N$(6)="NOURRITURE"
120 LET N$(7)="ASSURANCES"
130 LET N$(8)="DIVERS"
140 PRINT"TOTAL
    DES DEPENSES"
150 INPUT T
160 LET M=0
170 LET D=0
```

La deuxième partie du programme place dans le tableau D la valeur des dépenses saisies au clavier (instruction INPUT de la ligne 220). Il peut arriver que la somme des dépenses saisies (variable D) soit supérieure à la somme que vous pensiez avoir réellement dépensée (variable T). Dans ce cas, le message de la ligne 270 apparaît à l'écran et les dépenses diverses deviennent automatiquement nulles. Le tableau D constitué, on place dans le tableau P le pourcentage de chaque dépense par rapport à la variable M.

```
180 REM -------------------------
190 REM SAISIE DES DEPENSES
200 FOR I=1 TO 7
210 PRINT N$(I)
220 INPUT D(I)
230 IF D(I)<M THEN GOTO 250
240 LET M=D(I)
250 LET D=D+D(I)
260 IF D<=T THEN GOTO 290
270 PRINT "TOTAL > AUX DEPENSES"
280 LET T=D
290 NEXT I
300 LET D(8)=T-D
310 IF D(8)>M THEN LET M=D(8)
320 FOR I=1 TO 8
330 LET P(I)=INT((D(I)/(M/100))*.15)
340 NEXT I
```

TOTAL DES DÉPENSES
? 10 000
IMPOTS
? 25 000
LOYER
? 17 000
VOITURE
? 13 000
VACANCES
? 8 000
VETEMENTS
? 2 500
NOURRITURE
? 15 000
ASSURANCES
? 6 700

La troisième et dernière partie du programme affiche à l'écran le dessin de l'histogramme. On utilise une boucle FOR/NEXT pour passer en revue les huit dépenses et imprimer leurs libellés (tableau N$). Puis une seconde boucle imbriquée affiche les étoiles correspondant aux valeurs du tableau P.

Le nombre des dépenses est fixé à huit dans le programme, mais rien ne vous empêche d'en ajouter, d'en retrancher ou tout simplement de modifier les titres si vous n'avez pas de voiture ou si vous ne partez jamais en vacances.

```
350 REM ----------------------------
360 REM HISTOGRAMME
370 PRINT TAB(15);"I--------------->"
380 FOR I=1 TO 8
390 PRINT N$(I);TAB(15);"I";
400 FOR J=1 TO P(I)
410 PRINT"*";
420 NEXT J
430 PRINT
440 NEXT I
```



```
               I--------------->
IMPOTS         I***************
LOYER          I***********
VOITURE        I*******
VACANCES       I****
VETEMENTS      I*
NOURRITURE     I********
ASSURANCES     I****
DIVERS         I*****
```

BUDGET FAMILIAL - BUDGET F

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10 REM BUDGET FAMILIAL
20 REM ------------------------
30 DIM N$(8)
40 DIM D(8)
50 DIM P(8)
60 LET N$(1)="IMPOTS"
70 LET N$(2)="LOYER"
80 LET N$(3)="VOITURE"
90 LET N$(4)="VACANCES"
100 LET N$(5)="VETEMENTS"
110 LET N$(6)="NOURRITURE"
120 LET N$(7)="ASSURANCES"
130 LET N$(8)="DIVERS"
140 PRINT"TOTAL DES DEPENSES"
150 INPUT T
160 LET M=0
170 LET D=0
180 REM ------------------------
190 REM SAISIE DES DEPENSES
200 FOR I=1 TO 7
210 PRINT N$(I)
220 INPUT D(I)
230 IF D(I)<M THEN GOTO 250
240 LET M=D(I)
250 LET D=D+D(I)
260 IF D<=T THEN GOTO 290
270 PRINT "TOTAL > AUX DEPENSES"
280 LET T=D
290 NEXT I
300 LET D(8)=T-D
310 IF D(8)>M THEN LET M=D(8)
320 FOR I=1 TO 8
330 LET P(I)=INT((D(I)/(M/100))*.15)
340 NEXT I
350 REM --------------------------
360 REM HISTOGRAMME
370 PRINT TAB(15);"I--------------->"
380 FOR I=1 TO 8
390 PRINT N$(I);TAB(15);"I";
400 FOR J=1 TO P(I)
410 PRINT"*";
420 NEXT J
430 PRINT
440 NEXT I
```

Liste des variables

D	Dépenses totales (sauf divers)	**M**	Valeur maximale des dépenses
D()	Valeurs des dépenses	**N$(**	Libellés des dépenses
I	Indice de boucle	**P()**	Pourcentages des dépenses
J	Indice de boucle	**T**	Total des dépenses



TRICOT

A vos aiguilles ! Finis les longs calculs fastidieux pour connaître le nombre de mailles et de rangs à monter et à tricoter, en adaptant les modèles standard à la taille de chaque personne.

par Jean-François Sehan

Il suffit de donner à l'ordinateur le nombre de mailles et de rangs d'un échantillon de 10 cm sur 10 (variable M1 et R1). L'unité de calcul étant le centimètre, on divise par avance ces valeurs par 10.

```
10 REM          TRICOT
20 REM -----------------------------
30 PRINT "ECHANTILLON 10 CMS SUR 10
40 PRINT "NOMBRE DE MAILLES"
50 INPUT M1
60 PRINT "NOMBRE DE RANGS"
70 INPUT R1
80 LET M2=M1/10
90 LET R2=R1/10
```

Le programme vous demande toutes les coordonnées du pull à tricoter : largeur et hauteur du dos, du devant et des manches.

```
100 REM ------------------------------
110 REM MESURES DEVANT ET DOS
120 PRINT "LARGEUR DOS/DEVANT A LA
130 INPUT L1                   BASE"
140 PRINT "LARGEUR POITRINE"
150 INPUT L2
160 PRINT "HAUTEUR SOUS LES BRAS"
170 INPUT H1
180 PRINT "HAUTEUR TOTALE DOS/DEVANT"
190 INPUT H2
250 REM ------------------------------
260 REM MESURES MANCHES
270 PRINT "LARGEUR POIGNET"
280 INPUT P1
290 PRINT "LARGEUR MANCHE A L'EPAULE"
300 INPUT E1
310 PRINT "HAUTEUR MANCHE A L'EPAULE"
320 INPUT H3
330 PRINT "HAUTEUR TOTALE MANCHE"
340 INPUT H4
```

A la fin de chaque groupe de mesures, le programme les transforme en nombre de mailles et de rangs. Il calcule aussi le nombre de mailles des augmentations et des diminutions à effectuer de chaque côté du tricot.

```
200 LET N1=INT(M2*L1)
210 LET N2=INT(M2*L2)
220 LET N3=INT(R2*H1)
230 LET N4=INT(R2*H2)
240 LET D=INT((N1-N2)/2)
```

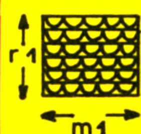
```
350 LET P2=INT(M2*P1)
360 LET E2=INT(M2*E1)
370 LET N5=INT(R2*H3)
380 LET N6=INT(R2*H4)
390 LET A=INT((E2-P2)/2)
```

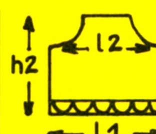
Les deux dernières parties du programme affichent à l'écran toutes les valeurs précédemment calculées. Elles affichent également le nombre de mailles qu'il restera à rabattre en fin de tricot.

```
400 REM --------------------------------
410 REM RESULTATS DOS ET DEVANT
420 PRINT " *-= MONTAGE DOS OU DEVANT =-*"
430 PRINT "-MONTER";N1;"MAILLES,SOIT";L1;"CMS."
440 PRINT "-TRICOTER";N3;"RANGS,SOIT";H1;"CMS."
450 PRINT "-A";H1;"CMS, RABATTRE DE CHAQUE COTE"
460 PRINT D;"MAILLES, EN PLUSIEURS FOIS,"
470 PRINT " POUR LES EMMANCHURES."
480 LET N7=N4-N3
490 LET C=H2-H1
500 PRINT "-TRICOTER ENCORE";N7;"RANGS,SOIT"
510 PRINT C;"CMS."
520 PRINT "-A";H2;"CMS,FERMER SELON L'ENCOLURE."
530 REM ----------------------------
540 REM RESULTATS MANCHES
550 PRINT " *-= MONTAGE DES MANCHES =-*"
560 PRINT "-MONTER";P2;"MAILLES,SOIT";P1;"CMS."
570 PRINT "-TRICOTER";N5;"RANGS,SOIT";H3;"CMS,"
580 PRINT " EN AUGMENTANT REGULIEREMENT"
590 PRINT " CHAQUE COTE DE";A;"MAILLES."
600 LET N8=(P2+(A*2))
610 LET N9=INT(N8/2)
620 PRINT "-A";H3;"CMS,RABATTRE LES";N8;"MAILLES"
630 PRINT " RESTANTES REGULIEREMENT,SOIT";N9
640 PRINT " MAILLES DE CHAQUE COTE."
650 PRINT "-A";H4;"CMS,FERMER LE TRICOT."
```

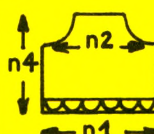
```
MONTAGE DOS OU DEVANT
MONTER 124 MAILLES, SOIT
48 CM
TRICOTER 136 RANGS, 40 CM
A 40 CM, RABATTRE DE CHAQUE
COTE 10 MAILLES, EN PLU-
SIEURS FOIS
POUR LES EMMANCHURES.
TRICOTER ENCORE 61 RANGS,
SOIT 18 CM
A 58 CM, FERMER SELON L'EN-
COLURE
```

```
MONTAGE DES MANCHES
MONTER 57 MAILLES, SOIT
22 CM
TRICOTER 146 RANGS, SOIT
43 CM EN AUGMENTANT REGU-
LIEREMENT CHAQUE COTE DE
18 MAILLES
A 43 CM, RABATTRE LES 93 MAIL-
LES RESTANTES REGULIERE-
MENT, SOIT 46 MAILLES DE CHA-
QUE COTE
A 56 CM, FERMER LE TRICOT
```











TRICOT-TRICOT-TRICOT-TRIC

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10 REM          TRICOT
20 REM ------------------------------
30 PRINT "ECHANTILLON 10 CM  SUR 10 CM "
40 PRINT "NOMBRE DE MAILLES"
50 INPUT M1
60 PRINT "NOMBRE DE RANGS"
70 INPUT R1
80 LET M2=M1/10
90 LET R2=R1/10
100 REM -----------------------------
110 REM MESURES DEVANT ET DOS
120 PRINT "LARGEUR DOS/DEVANT A LA BASE"
130 INPUT L1
140 PRINT "LARGEUR POITRINE"
150 INPUT L2
160 PRINT "HAUTEUR SOUS LES BRAS"
170 INPUT H1
180 PRINT "HAUTEUR TOTALE DOS/DEVANT"
190 INPUT H2
200 LET N1=INT(M2*L1)
210 LET N2=INT(M2*L2)
220 LET N3=INT(R2*H1)
230 LET N4=INT(R2*H2)
240 LET D=INT((N1-N2)/2)
250 REM -----------------------------
260 REM MESURES MANCHES
270 PRINT "LARGEUR POIGNET"
280 INPUT P1
290 PRINT "LARGEUR MANCHE A L'EPAULE"
300 INPUT E1
310 PRINT "HAUTEUR MANCHE A L'EPAULE"
320 INPUT H3
330 PRINT "HAUTEUR TOTALE MANCHE"
340 INPUT H4
350 LET P2=INT(M2*P1)
360 LET E2=INT(M2*E1)
370 LET N5=INT(R2*H3)
380 LET N6=INT(R2*H4)
390 LET A=INT((E2-P2)/2)
400 REM -----------------------------
410 REM RESULTATS DOS ET DEVANT
420 PRINT " *-= MONTAGE DOS OU DEVANT =-*"
430 PRINT "-MONTER";N1;"MAILLES,SOIT";L1;"CM  "
440 PRINT "-TRICOTER";N3;"RANGS,SOIT";H1;"CM  "
450 PRINT "-A";H1;"CM , RABATTRE DE CHAQUE COTE"
460 PRINT D;"MAILLES, EN PLUSIEURS FOIS,"
470 PRINT " POUR LES EMMANCHURES."
480 LET N7=N4-N3
490 LET C=H2-H1
500 PRINT "-TRICOTER ENCORE";N7;"RANGS,SOIT"
510 PRINT C;"CM  "
520 PRINT "-A";H2;"CM ,FERMER SELON L'ENCOLURE."
530 REM -----------------------------
540 REM RESULTATS MANCHES
550 PRINT " *-= MONTAGE DES MANCHES =-*"
560 PRINT "-MONTER";P2;"MAILLES,SOIT";P1;"CM  "
570 PRINT "-TRICOTER";N5;"RANGS,SOIT";H3;"CM  "
580 PRINT " EN AUGMENTANT REGULIEREMENT"
590 PRINT " CHAQUE COTE DE";A;"MAILLES."
600 LET N8=(P2+(A*2))
610 LET N9=INT(N8/2)
620 PRINT "-A";H3;"CMS,RABATTRE LES";N8;"MAILLES"
630 PRINT " RESTANTES REGULIEREMENT,SOIT";N9
640 PRINT " MAILLES DE CHAQUE COTE."
650 PRINT "-A";H4;"CMS,FERMER LE TRICOT."
```



MISSIONNAIRES ET CANNIBALES

Pourrez-vous faire traverser les trois missionnaires et les trois cannibales de l'autre côté de la rivière, compte tenu du fait que les missionnaires ne doivent être à aucun moment en minorité sur l'une des deux berges ?

C'est un jeu d'aventure... historique, bien entendu, puisque les cannibales ont disparu de la planète... de même que les missionnaires !

par Jean-François Sehan

Les trois missionnaires et les trois cannibales ont à leur disposition une barque pour traverser, mais celle-ci ne peut prendre que deux passagers au maximum. Il est donc nécessaire d'effectuer plusieurs passages pour que tout le monde se retrouve sur la berge de droite. A chaque passage le programme demande le nombre de cannibales et de missionnaires. Le total de ces deux nombres ne doit pas être nul ou supérieur à deux. Si les missionnaires deviennent minoritaires (2 cannibales contre 1 missionnaire, par exemple), vous perdez. A vous de trouver la bonne tactique !

Pour connaître le nombre des missionnaires et des cannibales à tout instant, on définit un tableau N( ) dans lequel on place les quantités de départ. La variante B permet de savoir sur quelle berge on se trouve (B=0 pour la berge de gauche, et B=1 pour celle de droite).

```
10 REM MISSIONNAIRES ET CANNIBALES
20 REM ---------------------------
30 DIM N(4)
40 LET N(1)=3
50 LET N(2)=3
60 LET B=0
```

Après avoir affiché le numéro de la berge, le programme demande le nombre de missionnaires et de cannibales qui doivent traverser, tout en vérifiant les impossibilités éventuelles.

```
70 GOSUB 360
80 PRINT
90 PRINT "VOUS ETES SUR LA BERGE";B+1
100 PRINT "NOMBRE DE CANNIBALES "
110 INPUT C
120 IF C>N(1+(B*2)) THEN GOTO 70
130 PRINT "NOMBRE DE MISSIONNAIRES "
140 INPUT M
150 IF M>N(2+(B*2)) THEN GOTO 70
160 IF C+M>2 THEN GOTO 70
170 IF C+M<1 THEN GOTO 70
```

```
BERGE 1            BERGE 2
CCC MMM

VOUS ETES SUR LA BERGE 1
NBRE DE CANNIBALES
? 1
NBRE DE MISSIONNAIRES
? 1
```

Les lignes 180 à 250 modifient le tableau N( ) en fonction des deux valeurs saisies au clavier (variable C et M) et changent le contenu de B pour la nouvelle berge.

```
180 N(1+(B*2))=N(1+(B*2))-C
190 N(2+(B*2))=N(2+(B*2))-M
200 IF B=1 THEN GOTO 230
210 LET B=1
220 GOTO 240
230 LET B=0
240 N(1+(B*2))=N(1+(B*2))+C
250 N(2+(B*2))=N(2+(B*2))+M
```

Avant de saisir au clavier de nouvelles données, on vérifie si tous les antagonistes ne sont pas arrivés à destination (vous avez gagné !) ou si les missionnaires ne sont pas en minorité (vous avez perdu !).

```
260 GOSUB 360
270 IF N(3)+N(4)=6 THEN GOTO 340
280 IF N(1)<=N(2) THEN GOTO 300
290 IF N(2)<>0 THEN GOTO 320
300 IF N(3)<=N(4) THEN GOTO 80
310 IF N(4)=0 THEN GOTO 80
320 PRINT "        VOUS AVEZ PERDU"
330 END
340 PRINT "        VOUS AVEZ GAGNE"
350 END
```

Le sous-programme des lignes 360 à 560 affiche à l'écran les deux berges et les lettres C pour les cannibales et M pour les missionnaires. On utilise un sous-programme car, deux fois dans le déroulement du programme, on a besoin de l'affichage du jeu complet (aux lignes 70 et 260).

```
360 REM ----------------------------
370 REM AFFICHE LE JEU
380 PRINT "----------------------------"
390 PRINT "BERGE 1                BERGE 2"
400 FOR I=1 TO 4
410 IF I/2=INT(I/2) THEN GOTO 440
420 LET C$="C"
430 GOTO 450
440 LET C$="M"
450 FOR J=1 TO 3
460 IF N(I)<J THEN GOTO 490
470 PRINT C$;
480 GOTO 500
490 PRINT " ";
500 NEXT J
510 IF I<>2 THEN GOTO 530
520 PRINT "           ";
530 PRINT " ";
540 NEXT I
550 PRINT
560 RETURN
```

```
BERGE 1            BERGE 2
CC MM                  C M

VOUS ETES SUR LA BERGE 2
NBRE DE CANNIBALES
? 0
NBRE DE MISSIONNAIRES
? 1

BERGE 1            BERGE 2
CC MMM             C
```

Liste des variables

N()	Nombre de cannibales et de missionnaires sur chaque berge
B	Numéro de la berge
C	Nombre de cannibales à déplacer
M	Nombre de missionnaires à déplacer
I	Indice de boucle FOR/NEXT
C$	Caractère à afficher
J	Indice de boucle FOR/NEXT

MISSIONNAIRES ET CANNIBALES–MISSIONN

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10 REM MISSIONNAIRES ET CANNIBALES
20 REM --------------------------
30 DIM N(4)
40 LET N(1)=3
50 LET N(2)=3
60 LET B=0
70 GOSUB 360
80 PRINT
90 PRINT "VOUS ETES SUR LA BERGE";B+1
100 PRINT "NOMBRE DE CANNIBALES "
110 INPUT C
120 IF C>N(1+(B*2)) THEN GOTO 70
130 PRINT "NOMBRE DE MISSIONNAIRES "
140 INPUT M
150 IF M>N(2+(B*2)) THEN GOTO 70
160 IF C+M>2 THEN GOTO 70
170 IF C+M<1 THEN GOTO 70
180 N(1+(B*2))=N(1+(B*2))-C
190 N(2+(B*2))=N(2+(B*2))-M
200 IF B=1 THEN GOTO 230
210 LET B=1
220 GOTO 240
230 LET B=0
240 N(1+(B*2))=N(1+(B*2))+C
250 N(2+(B*2))=N(2+(B*2))+M
260 GOSUB 360
270 IF N(3)+N(4)=6 THEN GOTO 340
280 IF N(1)<=N(2) THEN GOTO 300
290 IF N(2)<>0 THEN GOTO 320
300 IF N(3)<=N(4) THEN GOTO 80
310 IF N(4)=0 THEN GOTO 80
320 PRINT "      VOUS AVEZ PERDU"
330 END
340 PRINT "      VOUS AVEZ GAGNE"
350 END
360 REM ----------------------------
370 REM AFFICHE LE JEU
380 PRINT "----------------------------"
390 PRINT "BERGE 1                BERGE 2"
400 FOR I=1 TO 4
410 IF I/2=INT(I/2) THEN GOTO 440
420 LET C$="C"
430 GOTO 450
440 LET C$="M"
450 FOR J=1 TO 3
460 IF N(I)<J THEN GOTO 490
470 PRINT C$;
480 GOTO 500
490 PRINT " ";
500 NEXT J
510 IF I<>2 THEN GOTO 530
520 PRINT "              ";
530 PRINT " ";
540 NEXT I
550 PRINT
560 RETURN
```

LA FONCTION RND

La fonction RND du Basic est très importante dans l'élaboration de la plupart des jeux sur ordinateur. Mais les nombres qu'elle donne sont-ils vraiment aléatoires ?

par Jean-François Sehan

Pour tester cette fonction, le programme qui suit tire trente mille nombres aléatoires compris entre 1 et 10. Après totalisations, le programme affiche le nombre et le pourcentage obtenus tous les mille tirages, ainsi que l'écart entre le plus petit et le plus grand.

Pour trente mille, l'écart obtenu ne devrait pas être supérieur à 1 %.

Pour obtenir des nombres entiers de 1 à 10, on multiplie RND par 10. En effet, RND ne donne que des nombres entre 0 et 1 non compris. En multipliant par dix, on obtient des nombres entre 0 et 10 non compris. Il suffit alors d'ajouter 1 et de garder uniquement la partie entière (rôle de la fonction INT).

Les possesseurs de Sinclair ou de TI-99 devront taper RND et non RND(1) comme dans le listing. Pour plus de sécurité, vérifiez dans le manuel de votre machine la syntaxe exacte.

Le résultat étant placé dans la variable A, on incrémente la case correspondante du tableau R( ) (dimensionné à la ligne 30) à chaque tirage.

Dans le but de ne pas ralentir l'exécution du programme, on affiche les résultats intermédiaires tous les mille tirages uniquement. La fonction IF de la ligne 70 teste cette éventualité.

```
10 REM TEST DE LA FONCTION RND
20 REM ----------------------
30 DIM R(10)
40 FOR I=1 TO 30000
50 LET A=INT(RND(1)*10)+1
60 LET R(A)=R(A)+1
70 IF INT(I/1000)<>I/1000 THEN GOTO 340
```

Pour connaître la plus grande et la plus petite valeur du test, on initialise aux lignes 120 et 130 les variables P et G. P contient 100 car on trouvera toujours une valeur inférieure à 100 tandis que G contient 0 car il existe automatiquement une valeur supérieure à 0.

Les compteurs T1 et T2 cumulent le total de tirages et de pourcentages. Il est toujours égal à la variable I, mais il est possible que T2 ne soit pas égal à 100 à cause des calculs de la ligne 190.

```
80 REM ----------------------
90 REM AFFICHE LE RESULTAT
100 LET T1=0
110 LET T2=0
120 LET P=100
130 LET G=0
140 PRINT " ----------------------------"
150 PRINT "     APRES";I;"FONCTIONS RND"
160 PRINT " ----------------------------"
170 PRINT "LE NB"," EST SORTI"," SOIT"
```

APRES 15 000 FONCTIONS RND		
LE NB	EST SORTI	SOIT

La boucle des lignes 180 à 270 affiche à l'écran le contenu du tableau R( ) ainsi que les pourcentages correspondants. On stocke dans la variable G ou P les valeurs supérieures ou inférieures à leurs contenus actuels.

```
180 FOR J=1 TO 10
190 LET B=R(J)/(I/100)
200 LET T1=T1+R(J)
210 LET T2=T2+B
220 IF B>P THEN GOTO 240
230 LET P=B
240 IF B<G THEN GOTO 260
250 LET G=B
260 PRINT J,R(J);"FOIS",B;"%"
270 NEXT J
```

1	1 473 FOIS	9.82 %
2	1 483 FOIS	9.886666 %
3	1 500 FOIS	10 %
4	1 544 FOIS	10.29333 %
5	1 551 FOIS	10.34 %
6	1 498 FOIS	9.986667 %
7	1 461 FOIS	9.74 %
8	1 480 FOIS	9.866667 %
9	1 553 FOIS	10.35333 %
10	1 457 FOIS	9.713333 %

La dernière partie du programme affiche les totaux, les valeurs minimales et maximales rencontrées ainsi que leur différence (G – P).

```
280 PRINT ," ------"," ----------"
290 PRINT "TOTAL",T1,T2;"%"
300 PRINT "MINI :";P;"%"
310 PRINT "MAXI :";G;"%"
320 PRINT "ECART:";G-P;"%"
330 PRINT
340 NEXT I
350 END
```

TOTAL 15 000 100 %
MINI : 9.713333 %
MAXI : 10.35333 %
ECART : .6400003 %

Si vous voulez affiner les résultats, il est toujours possible d'augmenter la boucle I jusqu'à cent mille en modifiant la ligne 40 :

FOR I=1 TO 100000

LA FONCTION RND-LA FONCT

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10 REM TEST DE LA FONCTION RND
20 REM ----------------------
30 DIM R(10)
40 FOR I=1 TO 30000
50 LET A=INT(RND(1)*10)+1
60 LET R(A)=R(A)+1
70 IF INT(I/1000)<>I/1000 THEN GOTO 340
80 REM ----------------------
90 REM AFFICHE LE RESULTAT
100 LET T1=0
110 LET T2=0
120 LET P=100
130 LET G=0
140 PRINT " ------------------------------"
150 PRINT "    APRES";I;"FONCTIONS RND"
160 PRINT " ------------------------------"
170 PRINT "LE NB"," EST SORTI"," SOIT"
180 FOR J=1 TO 10
190 LET B=R(J)/(I/100)
200 LET T1=T1+R(J)
210 LET T2=T2+B
220 IF B>P THEN GOTO 240
230 LET P=B
240 IF B<G THEN GOTO 260
250 LET G=B
260 PRINT J,R(J);"FOIS",B;"%"
270 NEXT J
280 PRINT ," ------"," ---------"
290 PRINT "TOTAL",T1,T2;"%"
300 PRINT "MINI :";P;"%"
310 PRINT "MAXI :";G;"%"
320 PRINT "ECART:";G-P;"%"
330 PRINT
340 NEXT I
350 END
```

Liste des variables

A	Nombre aléatoire entre 1 et 10
B	Pourcentage
I	Indice de boucle FOR/NEXT
J	Indice de boucle FOR/NEXT
P	Pourcentage le plus petit
G	Pourcentage le plus grand
R( )	Total des tirages par nombre aléatoire
T1	Total des tirages
T2	Total des pourcentages



ALERTE A LA BOMBE

Une bombe a été déposée dans un immeuble de vingt étages. Saurez-vous la désamorcer en moins de cinquante secondes ?

Évidemment, on ne peut pas dire que l'intitulé du jeu soit du meilleur goût, mais l'humour noir n'a-t-il pas toujours été un recours pour survivre dans des situations parfois dramatiques ?

par Jean-François Sehan

Chaque étage comporte quatre cents appartements (20×20). Pour trouver l'emplacement de la bombe, il suffit de donner à votre détecteur trois coordonnées pour obtenir un des quatre messages (glacial, froid, chaud ou brûlant) et déterminer ainsi sa position.
Pour stocker les coordonnées de la bombe et celles proposées par le joueur, on définit aux lignes 30 et 40 deux tableaux numériques X( ) et Y( ).
Puis on place dans le tableau M$( ) les quatre messages possibles et dans X( ) des nombres aléatoires compris entre 1 et 20.
Le temps qui vous est imparti étant de cinquante secondes (ou essais), on attribue à la variable T cette même valeur.

```
10 REM ALERTE A LA BOMBE
20 REM ----------------------
30 DIM X(3)
40 DIM Y(3)
50 DIM M$(4)
60 LET M$(1)="GLACIAL"
70 LET M$(2)="FROID"
80 LET M$(3)="CHAUD"
90 LET M$(4)="BRULANT"
100 FOR I=1 TO 3
110 LET X(I)=INT(RND(1)*20)+1
120 NEXT I
130 LET T=50
```

Après avoir affiché le temps qu'il vous reste (variable T), le programme saisit au clavier vos trois coordonnées (en les plaçant dans le tableau Y) et vérifie qu'elles ne soient ni inférieures à zéro ni supérieures à vingt.

```
160 REM --------------------
170 REM LE JEU
180 LET S=0
190 LET R=0
200 PRINT "IL VOUS RESTE";T;
                  "SECONDES"
210 FOR I=1 TO 3
220 PRINT"COORDONNEE";I
230 INPUT Y(I)
240 IF Y(I)>20 THEN GOTO 220
250 IF Y(I)<1 THEN GOTO 220
```

IL VOUS RESTE 50 SECONDES
COORDONNEE 1
? 10
COORDONNEE 2
? 10
COORDONNEE 3
? 10

Si une des coordonnées est correcte, on incrémente le compteur S. Dans le cas où elle se trouve à moins de six cases de la bombe, on incrémente alors le compteur R.

```
260 IF Y(I)<>X(I) THEN GOTO 280
270 LET S=S+1
280 IF Y(I)<X(I)-5 THEN GOTO 310
290 IF Y(I)>X(I)+5 THEN GOTO 310
300 LET R=R+1
310 NEXT I
```

Si le compteur S contient 3 après la boucle, le programme va en ligne 440 pour afficher le message « OUF ! » (fin du jeu).
Dans le cas contraire, on affiche un des quatre messages de M$ suivant la valeur de R, ainsi que le nombre de coordonnées justes.

```
320 IF S=3 THEN GOTO 440
330 LET R=R+1
340 PRINT "        *-=";M$(R);"=-*"
350 PRINT S;"COORDONNEE(S) JUSTE(S)
```

FROID
0 COORDONNEE(S) JUSTE(S)

Avant de retourner à une nouvelle saisie, on teste le contenu de la variable T pour vérifier qu'il vous reste du temps.

```
360 PRINT
370 LET T=T-1
380 IF T>0 THEN GOTO 180
```

Si ce n'est plus le cas, on affiche le message « BOUM ! » et les coordonnées qu'il fallait trouver.

```
390 REM ---------------------------
400 REM PERDU OU GAGNE ?
410 PRINT "BOUM ! A LA PROCHAINE ...."
420 PRINT "LA BOMBE ETAIT EN";X(1);
                          X(2);X(3)
430 END
440 PRINT "OUF! VOUS L'AVEZ TROUVEE"
450 END
```

IL VOUS RESTE 49 SECONDES
COORDONNEE 1
? 5
COORDONNEE 2
? 10
COORDONNEE 3
? 15
BRULANT
0 COORDONNEE(S) JUSTE(S)

ALERTE A LA BOMBE - ALERTE A LA BOMBE

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10 REM ALERTE A LA BOMBE
20 REM ------------------------------
30 DIM X(3)
40 DIM Y(3)
50 DIM M$(4)
60 LET M$(1)="GLACIAL"
70 LET M$(2)="FROID"
80 LET M$(3)="CHAUD"
90 LET M$(4)="BRULANT"
100 FOR I=1 TO 3
110 LET X(I)=INT(RND(1)*20)+1
120 NEXT I
130 LET T=50
140 PRINT "LA BOMBE EST DANS UNE DES 8000 PIECES"
150 PRINT "VOUS AVEZ 50 SECONDES POUR LA TROUVER"
160 REM ------------------------------
170 REM LE JEU
180 LET S=0
190 LET R=0
200 PRINT "IL VOUS RESTE";T;"SECONDES"
210 FOR I=1 TO 3
220 PRINT"COORDONNEE";I
230 INPUT Y(I)
240 IF Y(I)>20 THEN GOTO 220
250 IF Y(I)<1 THEN GOTO 220
260 IF Y(I)<>X(I) THEN GOTO 280
270 LET S=S+1
280 IF Y(I)<X(I)-5 THEN GOTO 310
290 IF Y(I)>X(I)+5 THEN GOTO 310
300 LET R=R+1
310 NEXT I
320 IF S=3 THEN GOTO 440
330 LET R=R+1
340 PRINT "      *-=";M$(R);"=-*"
350 PRINT S;"COORDONNEE(S) JUSTE(S)"
360 PRINT
370 LET T=T-1
380 IF T>0 THEN GOTO 180
390 REM ------------------------------
400 REM PERDU OU GAGNE ?
410 PRINT "BOUM ! A LA PROCHAINE ...."
420 PRINT "LA BOMBE ETAIT EN";X(1);X(2);X(3)
430 END
440 PRINT "OUF! VOUS L'AVEZ TROUVEE"
450 END
```

Liste des variables

X( )	Coordonnées de la bombe	T	Temps restant
Y( )	Coordonnées proposées	S	Nombre de coordonnées correctes
M$( )	Messages	R	Nombre de coordonnées proches
I	Indice de boucle FOR/NEXT		



PRONOSTICS

Comment obtenir un tiercé unique en se référant à plusieurs pronostics.

par Jean-François Sehan

Le programme proposé donne une note pour chaque cheval, en fonction de la place attribuée par chaque pronostiqueur. On trie ces notes pour obtenir un classement, comme on le fait avec les notes des élèves pour connaître le premier de la classe, le deuxième, etc.
Pour vous permettre de faire votre choix, le programme affiche non seulement la liste des chevaux avec leurs notes, mais suggère aussi quatre tiercés différents, composés des premiers chevaux de la liste précédente.
Mais parlons plutôt du programme. Les pronostics sont placés dans des tableaux. Il faut donc préciser au Basic le nom et la grandeur de ceux-ci. C'est le rôle des instructions DIM des lignes 30 et 40. La taille maximale (le nombre de chevaux, bien sûr) est fixée à 30, comme on peut le vérifier sur les tickets de PMU.

```
10 REM PRONOSTICS
20 REM ----------------------
30 DIM T(30)
40 DIM R(30)
```

Pour mener à bien l'opération de classement, le programme a besoin des données suivantes :
* nombre de chevaux en course : pour être plus rapide dans nos calculs, on ne tient compte que du nombre de partants.
* nombre de pronostics : donnez simplement le nombre de pronostics en votre possession (journaux, radio ou télévision).
* nombre de chevaux par pronostics : les pronostiqueurs donnent leurs tiercés en 5 ou 6 chevaux, voire 8. Vous devez taper au clavier, en réponse à cette question, le nombre de chevaux du pronostiqueur qui en propose le moins.

```
50 PRINT"NOMBRE DE CHEVAUX
        EN COURSE "
60 INPUT N
70 FOR I=1 TO N
80 LET R(I)=I
90 NEXT I
100 PRINT"NBRE DE PRONOSTICS"
110 INPUT P
120 PRINT "NOMBRE DE CHEVAUX
          PAR PRONOSTIC "
130 INPUT C
```

Puis c'est le tour des pronostics eux-mêmes. Le programme affiche le numéro du pronostic et vous demande de taper au clavier les numéros des chevaux les uns à la suite des autres. On place alors dans le tableau T le nombre de points. S'il y a trois chevaux par pronostic, le premier obtient 3 points, le second 2 points et le dernier un seul point.

```
140 FOR I=1 TO P
150 PRINT "PRONOSTIC No";I
160 FOR J=1 TO C
170 PRINT "No DU CHEVAL ";J
180 INPUT A
190 LET T(A)=T(A)+C-J+1
200 NEXT J
210 PRINT
220 NEXT I
```

La deuxième partie du programme classe le tableau T par ordre croissant de points. On compare pour cela la case 1 avec la case 2. Si la case 2 est inférieure, on échange alors le contenu des deux cases (idem pour le tableau R) et on met le drapeau à 1 (variable D=1 en ligne 340). Puis on compare la case 2 avec la case 3, la 3 avec la 4, etc. Si l'on a effectué au moins une inversion (le drapeau D est à 1), on recommence le classement, et ceci tant que le drapeau D n'est pas à 0.

```
NOMBRE DE CHEVAUX EN COURSE
? 25
NOMBRE DE PRONOSTICS
? 4
NOMBRE DE CHEVAUX PAR PRONOSTIC
? 5
PRONOSTIC N° 1
N° DU CHEVAL 1
? 5
N° DU CHEVAL 2
? 23
N° DU CHEVAL 3
? ... etc...
```

```
230 REM ------------------
240 REM CLASSEMENT
250 LET D=0
260 FOR I=1 TO N-1
270 IF T(I)>=T(I+1) THEN 350
280 LET B=T(I)
290 LET T(I)=T(I+1)
300 LET T(I+1)=B
310 LET B=R(I)
320 LET R(I)=R(I+1)
330 LET R(I+1)=B
340 LET D=1
350 NEXT I
360 IF D=1 THEN 250
```

La dernière partie du programme affiche les deux tableaux R et T (numéro de cheval et nombre de points) dans l'ordre où ils ont été classés. Les lignes 450 à 540 vous proposent quatre combinaisons possibles de tiercé.

```
370 REM -------------------
380 REM RESULTATS
390 PRINT "CHEVAUX","POINTS"
400 FOR I=1 TO N
410 IF T(I)=0 THEN 430
420 PRINT R(I),T(I)
430 NEXT I
440 PRINT
450 PRINT "VOUS POUVEZ JOUER"
460 FOR I=1 TO 4
470 FOR J=0 TO 2
480 IF J+I<=4 THEN 510
490 PRINT R(J+I-4),
500 GOTO 520
510 PRINT R(J+I),
520 NEXT J
530 PRINT
540 NEXT I
550 END
```

```
CHEVAUX   POINTS
  25        10
  15         8
  12         7
   6         6
  23         5
   3         3
   8         3
   9         3
VOUS POUVEZ JOUER
  25        15        12
  15        12         6
  12         6        25
   6        25        15
```

PRONOSTICS - PRONOSTICS -

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10 REM PRONOSTICS
20 REM -----------------------------------
30 DIM T(30)
40 DIM R(30)
50 PRINT"NOMBRE DE CHEVAUX EN COURSE "
60 INPUT N
70 FOR I=1 TO N
80 LET R(I)=I
90 NEXT I
100 PRINT"NOMBRE DE PRONOSTICS "
110 INPUT P
120 PRINT "NOMBRE DE CHEVAUX PAR PRONOSTIC "
130 INPUT C
140 FOR I=1 TO P
150 PRINT "PRONOSTIC No";I
160 FOR J=1 TO C
170 PRINT "No DU CHEVAL ";J
180 INPUT A
190 LET T(A)=T(A)+C-J+1
200 NEXT J
210 PRINT
220 NEXT I
230 REM ------------------
240 REM CLASSEMENT
250 LET D=0
260 FOR I=1 TO N-1
270 IF T(I)>=T(I+1) THEN 350
280 LET B=T(I)
290 LET T(I)=T(I+1)
300 LET T(I+1)=B
310 LET B=R(I)
320 LET R(I)=R(I+1)
330 LET R(I+1)=B
340 LET D=1
350 NEXT I
360 IF D=1 THEN 250
370 REM --------------------
380 REM RESULTATS
390 PRINT "CHEVAUX","POINTS"
400 FOR I=1 TO N
410 IF T(I)=0 THEN 430
420 PRINT R(I),T(I)
430 NEXT I
440 PRINT
450 PRINT "VOUS POUVEZ JOUER"
460 FOR I=1 TO 4
470 FOR J=0 TO 2
480 IF J+I<=4 THEN 510
490 PRINT R(J+I-4),
500 GOTO 520
510 PRINT R(J+I),
520 NEXT J
530 PRINT
540 NEXT I
550 END
```

Liste des variables

A	numéro de cheval dans un pronostic
B	utilisée pour le classement
C	nombre de chevaux par pronostic
D	drapeau ; D = 1 si le classement est incomplet
I	indice de boucle FOR/NEXT
J	indice de boucle FOR/NEXT
N	nombre de chevaux en course
P	nombre de pronostics
R()	numéro de chaque cheval
T()	total des points de chaque cheval



JOUEZ AU COMPTE EST BON

On ne compte plus les adeptes du célèbre jeu télévisé « Le compte est bon ». Manque d'entraînement ? Votre ordinateur familial doté de ce programme original vous sera d'un grand secours.

par Jean-François Sehan

Si le résultat est le même que celui du jeu télévisé, la démarche pour obtenir la solution est tout autre. En effet, le problème est pris à l'envers. Le programme ne recherche pas une solution pour un nombre donné, mais effectue une suite de calculs avec des nombres et des opérations pris au hasard pour obtenir le nombre que vous devrez trouver. Rechercher une solution avec le raisonnement d'un joueur est une tâche beaucoup plus compliquée, qui mériterait un programme bien plus « lourd ».

Dès le lancement, l'ordinateur affiche à l'écran les sept nombres tirés au hasard et le nombre qu'il faut trouver, calculé par combinaison des sept nombres précédents. Quand vous avez une solution, appuyez sur la touche ENTER (RETURN ou NEW LINE pour certains ordinateurs) pour obtenir celle du programme. Toutes les opérations devant être conservées, on dimensionne quatre tableaux à cet effet (lignes 30 à 60). Les chiffres de départ sont placés dans le tableau C().

```
10 REM LE COMPTE EST BON
20 REM --------------------
30 DIM C(7)
40 DIM S(7)
50 DIM O(7)
60 DIM T(7)
```

« Le compte est bon » utilise les chiffres de 1 à 10 mais aussi les nombres 25, 50, 75 et 100.

La fonction RND de la ligne 90 donne un nombre compris entre 1 et 14. Si celui-ci est supérieur à 10, on en retranche 10 et on le multiplie par 25 pour obtenir les quatre nombres supérieurs à 10 (RND(1) s'écrit RND pour le TI99 et le ZX 81/Spectrum).

```
70 PRINT "JE PROPOSE :"
80 LET I=1
90 LET C(I)=INT(RND(1)*14)+1
100 IF C(I)<11 THEN 120
110 LET C(I)=(C(I)-10)*25
120 IF I>1 THEN 150
130 LET R=C(I)
140 GOTO 290
```

La fonction RND de la ligne 150 donne un chiffre entre 1 et 4 pour le type d'opération à effectuer. Après calcul, le résultat est stocké dans la variable R. Cette dernière est vérifiée en ligne 290 et 300, car le résultat doit rester dans la fourchette de 1 à 999. Si l'opération est correcte, le nombre choisi et le total sont placés respectivement dans les tableaux S() et T().

Quand les sept opérations ont été effectuées, on vérifie en dernier lieu que le résultat obtenu est supérieur à 100 (règle du jeu).

```
150 LET O(I)=INT(RND(1)*4)+1
160 IF O(I)=1 THEN 210
170 IF O(I)=2 THEN 230
180 IF O(I)=3 THEN 260
190 LET R=R+C(I)
200 GOTO 290
210 LET R=R-C(I)
220 GOTO 290
230 IF C(I)=1 THEN 80
240 LET R=R*C(I)
250 GOTO 290
260 IF C(I)=1 THEN 80
270 IF INT(R/C(I))<>R/C(I)THEN 80
280 LET R=R/C(I)
290 IF R>999 THEN 80
300 IF R<1 THEN 80
310 LET S(I)=C(I)
320 LET T(I)=R
330 LET I=I+1
340 IF I<8 THEN 90
350 IF R<100 THEN 80
```

Pour tromper le joueur, on affiche de façon aléatoire le tableau C(). Dès qu'un des éléments de ce tableau a été choisi par la fonction RND de la ligne 390, on met celle-ci à zéro (ligne 420).

```
360 REM --------------------
370 REM MELANGE/EDITION
380 FOR I=1 TO 7
390 LET A=INT(RND(1)*7)+1
400 IF C(A)=0 THEN 390
410 PRINT C(A);
420 LET C(A)=0
430 NEXT I
440 PRINT
450 PRINT "POUR TROUVER: ";R
460 PRINT"TAPEZ SUR UNE TOUCHE"
470 INPUT Z$
```

JE PROPOSE :
75 25 100 9 100 10 75
POUR TROUVER : 142
TAPEZ SUR UNE TOUCHE

La troisième partie du programme affiche à l'écran la solution à l'aide des tableaux S, O et T.

```
480 REM --------------------
490 REM SOLUTION
500 PRINT "ON COMMENCE AVEC";S(1)
510 FOR I=2 TO 7
520 PRINT T(I-1);
530 IF O(I)=1 THEN 580
540 IF O(I)=2 THEN 600
550 IF O(I)=3 THEN 620
560 PRINT "+";
570 GOTO 630
580 PRINT"-";
590 GOTO 630
600 PRINT"*";
610 GOTO 630
620 PRINT"/";
630 PRINT S(I);"=";T(I)
640 NEXT I
```

ON COMMENCE AVEC 9
9 * 75 = 675
675 / 25 = 27
27 − 10 = 17
17 + 100 = 117
117 − 75 = 42
42 + 100 = 142

JOUEZ AU COMPTE EST BON - JOUEZ A

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10 REM LE COMPTE EST BON
20 REM ------------------------
30 DIM C(7)
40 DIM S(7)
50 DIM O(7)
60 DIM T(7)
70 PRINT "JE PROPOSE :"
80 LET I=1
90 LET C(I)=INT(RND(1)*14)+1
100 IF C(I)<11 THEN 120
110 LET C(I)=(C(I)-10)*25
120 IF I>1 THEN 150
130 LET R=C(I)
140 GOTO 290
150 LET O(I)=INT(RND(1)*4)+1
160 IF O(I)=1 THEN 210
170 IF O(I)=2 THEN 230
180 IF O(I)=3 THEN 260
190 LET R=R+C(I)
200 GOTO 290
210 LET R=R-C(I)
220 GOTO 290
230 IF C(I)=1 THEN 80
240 LET R=R*C(I)
250 GOTO 290
260 IF C(I)=1 THEN 80
270 IF INT(R/C(I))<>R/C(I) THEN 80
280 LET R=R/C(I)
290 IF R>999 THEN 80
300 IF R<1 THEN 80
310 LET S(I)=C(I)
320 LET T(I)=R
330 LET I=I+1
340 IF I<8 THEN 90
350 IF R<100 THEN 80
360 REM ------------------------
370 REM MELANGE/EDITION
380 FOR I=1 TO 7
390 LET A=INT(RND(1)*7)+1
400 IF C(A)=0 THEN 390
410 PRINT C(A);
420 LET C(A)=0
430 NEXT I
440 PRINT
450 PRINT "POUR TROUVER: ";R
460 PRINT"TAPEZ SUR UNE TOUCHE"
470 INPUT Z$
480 REM ------------------------
490 REM SOLUTION
500 PRINT "ON COMMENCE AVEC";S(1)
510 FOR I=2 TO 7
520 PRINT T(I-1);
530 IF O(I)=1 THEN 580
540 IF O(I)=2 THEN 600
550 IF O(I)=3 THEN 620
560 PRINT "+";
570 GOTO 630
580 PRINT"-";
590 GOTO 630
600 PRINT"*";
610 GOTO 630
620 PRINT"/";
630 PRINT S(I);"=";T(I)
640 NEXT I
```

Liste des variables

A	utilisée pour le mélange
C()	liste des chiffres proposés
I	indice de boucle FOR/NEXT
O()	liste des opérations effectuées
R	résultat d'une opération
S()	liste des solutions
T()	total de chaque opération
Z$	variable du INPUT d'attente qu'une touche soit enfoncée



CUISINE

Pour les gourmets. Votre Ordinateur, toujours pratique, contribue à l'amélioration du quotidien en proposant cette fiche-cuisine. Elle vous permettra de réaliser de somptueux gâteaux.

par Jean-François Sehan

Toutes les données concernant les ingrédients sont placées dans les tableaux dimensionnés aux lignes 30 à 80. Ces données sont placées dans des instructions DATA. Si votre ordinateur ne possède pas cette instruction, vous devez remplacer les lignes 90 à 200 par :

```
 90 LET C$(1)="FARINE"
100 LET Q(1)=100
110 LET U(1)=2
120 LET A(1)=3
130 LET C$(2)="SUCRE"
140 LET Q(2)=50
150 LET U(2)=2
160 LET A(2)=2
170 LET C$(3)="LAIT"
    etc ... jusqu'à
880 LET A(20)=2
et pour le tableau des unités:
890 LET U$(1)=""
900 LET U$(2)="GRAMMES"
910 LET U$(3)="CUILLERE(S)"
920 LET U$(4)="CLS"
    etc ... jusqu'à
950 LET U$(7)="PINCEE(S)"
```

Les possesseurs de ZX81 doivent remplacer les lignes 30 et 70 par :

```
30 DIM C$(20,10)
70 DIM U$(7,12)
```

```
10 REM CUISINE
20 REM -----------------------------
30 DIM C$(20)
40 DIM Q(20)
50 DIM U(20)
60 DIM A(20)
70 DIM U$(7)
80 DIM G(7)
90 DATA FARINE,100,2,3,SUCRE,50,2,2,
        LAIT,10,4,2,EAU,10,4,2,OEUF,1,1,3
100 DATA VANILLE,.5,5,2,HUILE,1,3,2,
        BEURRE,50,2,2,LEVURE,.5,5,2
110 DATA BEURRE,50,2,2,RAISINS,50,2,2,
        RHUM,.5,6,1,NOIX,50,2,2
120 DATA CHOCOLAT,50,2,2,POMMES,200,2,2,
        SEL,1,7,2,CREME,25,2,2
130 DATA AMANDES,50,2,2,FRUITS C.,50,2,2,
        NOISETTES,50,2,2
140 DATA ,GRAMMES,CUILLERE(S),CLS,
        SACHET,VERRE,PINCEE(S)
150 FOR I=1 TO 20
160 READ C$(I),Q(I),U(I),A(I)
170 NEXT I
180 FOR I=1 TO 7
190 READ U$(I)
200 NEXT I
```

Le programme prend toujours comme base de départ pour la pâte les trois premiers ingrédients (farine, sucre et lait). Les lignes 1010 à 1060 font appel au sous-programme en 1250 pour obtenir une quantité aléatoire. Le tableau Q donne la quantité de base (100 grammes pour la farine par exemple) et le tableau A donne le multiplicateur maximum (3 pour la farine). Ainsi, la quantité de farine sera comprise entre 100 et 300 grammes.

On effectue la même opération pour les autres ingrédients. Le nombre d'éléments supplémentaires est donné par la variable S (nombre aléatoire entre 4 et 6). La boucle FOR/NEXT des lignes 1110 à 1130 vérifie dans le tableau G les éléments déjà sortis pour éviter les doublons. Comme pour les autres fiches programmes, il faut remplacer les instructions RND(1) par RND pour le TI99/4A et les ordinateurs Sinclair.

```
1000 PRINT "RECETTE"
1010 LET N=1
1020 GOSUB 1250
1030 LET N=2
1040 GOSUB 1250
1050 LET N=3
1060 GOSUB 1250
1070 LET S=INT(RND(1)*3)+4
1080 FOR I=1 TO S
1090 LET N=INT(RND(1)*17)+4
1100 IF I=1 THEN 1140
1110 FOR J=1 TO I-1
1120 IF N=G(J) THEN 1090
1130 NEXT J
1140 GOSUB 1250
1150 LET G(I)=N
1160 NEXT I
```

Les lignes 1170 à 1240 affichent à l'écran quelques consignes pour la préparation de votre gâteau.

Remarque : pour une utilisation réelle, il est préférable de vérifier que le mélange des ingrédients proposés est vraisemblable, cela évitera les indigestions...

```
1170 PRINT "MELANGER FARINE, SUCRE ET LAIT
1180 PRINT "POUR OBTENIR LA PATE."
1190 PRINT "AJOUTER LES AUTRES INGREDIENTS.
1200 PRINT "METTRE LA PATE OBTENUE DANS"
1210 PRINT "DANS UN MOULE BEURRE."
1220 PRINT "PLACER LE MOULE AU FOUR"
1230 PRINT "PENDANT 30 MINUTES."
1240 END
1250 PRINT C$(N);"  :";Q(N)*INT(RND(1)
                   *A(N)+1);U$(U(N))
1260 RETURN
```

RECETTE
FARINE : 100 GRAMMES
SUCRE : 100 GRAMMES
LAIT : 20 CLS
CRÈME : 25 GRAMMES
RHUM : 5 VERRES
ŒUF : 3
POMMES : 400 GRAMMES
NOISETTES : 50 GRAMMES
FRUITS C. : 100 GRAMMES
MÉLANGER FARINE, SUCRE ET LAIT POUR OBTENIR LA PÂTE.
AJOUTER LES AUTRES INGRÉDIENTS.
METTRE LA PÂTE OBTENUE DANS UN MOULE BEURRE.
PLACER LE MOULE AU FOUR PENDANT 30 minutes.

Liste des variables

- **A()** liste des multiplicateurs aléatoires
- **C$()** liste des composants
- **G()** liste des éléments déjà utilisés
- **I** indice de boucle FOR/NEXT
- **J** indice de boucle FOR/NEXT
- **N** numéro d'élément
- **Q()** liste des quantités de base
- **S** nombre d'éléments supplémentaires
- **U()** liste des unités
- **U$()** liste des noms des unités

CUISINE - CUISINE - CUISINE

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10 REM CUISINE
20 REM ------------------------------
30 DIM C$(20)
40 DIM Q(20)
50 DIM U(20)
60 DIM A(20)
70 DIM U$(7)
80 DIM G(7)
90 DATA FARINE,100,2,3,SUCRE,50,2,2,
        LAIT,10,4,2,EAU,10,4,2,OEUF,1,1,3
100 DATA VANILLE,.5,5,2,HUILE,1,3,2,
         BEURRE,50,2,2,LEVURE,.5,5,2
110 DATA BEURRE,50,2,2,RAISINS,50,2,2,
         RHUM,.5,6,1,NOIX,50,2,2
120 DATA CHOCOLAT,50,2,2,POMMES,200,2,2,
         SEL,1,7,2,CREME,25,2,2
130 DATA AMANDES,50,2,2,FRUITS C.,50,2,2,
         NOISETTES,50,2,2
140 DATA ,GRAMMES,CUILLERE(S),CLS,
         SACHET,VERRE,PINCEE(S)
150 FOR I=1 TO 20
160 READ C$(I),Q(I),U(I),A(I)
170 NEXT I
180 FOR I=1 TO 7
190 READ U$(I)
200 NEXT I
1000 PRINT "RECETTE"
1010 LET N=1
1020 GOSUB 1250
1030 LET N=2
1040 GOSUB 1250
1050 LET N=3
1060 GOSUB 1250
1070 LET S=INT(RND(1)*3)+4
1080 FOR I=1 TO S
1090 LET N=INT(RND(1)*17)+4
1100 IF I=1 THEN 1140
1110 FOR J=1 TO I-1
1120 IF N=G(J) THEN 1090
1130 NEXT J
1140 GOSUB 1250
1150 LET G(I)=N
1160 NEXT I
1170 PRINT "MELANGER FARINE, SUCRE ET LAIT"
1180 PRINT "POUR OBTENIR LA PATE."
1190 PRINT "AJOUTER LES AUTRES INGREDIENTS."
1200 PRINT "METTRE LA PATE OBTENUE DANS"
1210 PRINT "DANS UN MOULE BEURRE."
1220 PRINT "PLACER LE MOULE AU FOUR"
1230 PRINT "PENDANT 30 MINUTES."
1240 END
1250 PRINT C$(N);"  :";Q(N)*INT(RND(1)*A(N)+1);U$
1260 RETURN                                  (U(N))
```



LOTO

Si vous ne voulez plus jouer la date de naissance de votre concierge, si vous manquez d'idées pour remplir les grilles de Loto, laissez ce soin à votre ordinateur.

par Jean-François Sehan

Ce programme n'oublie pas les adeptes des grilles multiples. Dès son lancement, il vous demande d'entrer au clavier le nombre de chiffres désiré. La réponse est placée dans la variable M, après vérification.

```
NOMBRE DE CHIFFRES À TIRER
? 7
GRILLE MULTIPLE : 7 NUMÉROS
LE 3
LE 10
LE 11
LE 12
LE 30
LE 32
LE 49
UNE AUTRE GRILLE (O/N)
? 0
```

Les chiffres du tirage étant conservés dans le tableau N(), on dimensionne celui-ci en ligne 30 (instruction DIM).

```
10 REM LOTO
20 REM ---------------
30 DIM N(10)
```

Pour remplir ce tableau N() avec des nombres aléatoires, on utilise deux boucles FOR/NEXT imbriquées. La première, allant de 1 à M, place dans la variable A un nombre aléatoire compris entre 1 et 49. La deuxième boucle vérifie, dans le tableau N, si le nouveau nombre n'a pas déjà été tiré (ce test n'est pas effectué pour N(1) bien entendu). Si c'est le cas, on met le drapeau D à 1. A la fin de cette boucle (variable J), on vérifie le contenu de D. Si celui-ci est égal à 1, le nombre stocké dans A est déjà sorti, on retourne alors à la ligne 90 pour un nouveau tirage. Dans le cas contraire, on place le contenu de A dans le tableau N.

```
40 PRINT "NOMBRE DE CHIFFRES A TIRER"
50 INPUT M
60 IF M<6 THEN 40
70 IF M>10 THEN 40
80 FOR I=1 TO M
90 LET A=INT(RND(1)*49)+1
100 IF I=1 THEN 170
110 LET D=0
120 FOR J=1 TO I-1
130 IF A<>N(J) THEN 150
140 LET D=1
150 NEXT J
160 IF D=1 THEN 90
170 LET N(I)=A
180 NEXT I
```

```
NOMBRE DE CHIFFRES Á TIRER
? 7
```

Pour que les nombres soient plus facilement utilisables, on classe le tableau N dans l'ordre croissant. Pour cela, on compare la case 1 avec la case 2. Si la case 2 est inférieure, on échange les contenus des deux cases. Puis on compare la 2 avec la 3, la 3 avec la 4, etc.

	AVANT N()		APRÈS N()
1	39	1	24
2	24	2	39
3	6	3	6
4	28	4	28
5	43	5	43
6	49	6	49

On compare la case 1 avec la 2

Puis la 2 avec la 3, etc.

Si l'on a effectué au moins une inversion (le drapeau D est à 1), on recommence le classement et ceci tant que le drapeau n'est pas à 0.

```
190 REM -----------------------
200 REM CLASSEMENT
210 LET D=0
220 FOR I=1 TO M-1
230 IF N(I)<N(I+1) THEN 280
240 LET A=N(I)
250 LET N(I)=N(I+1)
260 LET N(I+1)=A
270 LET D=1
280 NEXT I
290 IF D=1 THEN 210
```

La dernière partie du programme édite à l'écran les nombres choisis pour une grille simple ou multiple.

```
300 REM ---------------------
310 REM EDITION
320 PRINT
330 IF M>6 THEN 360
340 PRINT "GRILLE SIMPLE: 6 NUMEROS"
350 GOTO 370
360 PRINT "GRILLE MULTIPLE:";M;"NUMEROS"
370 FOR I=1 TO M
380 PRINT "LE ";N(I)
390 NEXT I
400 PRINT
410 PRINT"UNE AUTRE GRILLE (O/N) "
420 INPUT R$
430 IF R$="O" THEN 80
440 END
```

```
GRILLE MULTIPLE :
7 NUMÉROS
LE 4
LE 6
LE 25
LE 32
LE 36
LE 40
LE 43
UNE AUTRE GRILLE (O/N)
? 0
```

VOTRE ORDINATEUR 14

FICHE PROGRAMME

LOTO - LOTO - LOTO - LOTO

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10 REM LOTO
20 REM -----------------------
30 DIM N(10)
40 PRINT "NOMBRE DE CHIFFRES A TIRER"
50 INPUT M
60 IF M<6 THEN 40
70 IF M>10 THEN 40
80 FOR I=1 TO M
90 LET A=INT(RND(1)*49)+1
100 IF I=1 THEN 170
110 LET D=0
120 FOR J=1 TO I-1
130 IF A<>N(J) THEN 150
140 LET D=1
150 NEXT J
160 IF D=1 THEN 90
170 LET N(I)=A
180 NEXT I
190 REM -----------------------
200 REM CLASSEMENT
210 LET D=0
220 FOR I=1 TO M-1
230 IF N(I)<N(I+1) THEN 280
240 LET A=N(I)
250 LET N(I)=N(I+1)
260 LET N(I+1)=A
270 LET D=1
280 NEXT I
290 IF D=1 THEN 210
300 REM ----------------------
310 REM EDITION
320 PRINT
330 IF M>6 THEN 360
340 PRINT "GRILLE SIMPLE: 6 NUMEROS"
350 GOTO 370
360 PRINT "GRILLE MULTIPLE:";M;"NUMEROS"
370 FOR I=1 TO M
380 PRINT "LE ";N(I)
390 NEXT I
400 PRINT
410 PRINT"UNE AUTRE GRILLE (O/N) "
420 INPUT R$
430 IF R$="O" THEN 80
440 END
```

Liste des variables

A	nombre tiré au hasard
D	drapeau ; si D = 1 nombre déjà tiré
I	indice de boucle FOR/NEXT
J	indice de boucle FOR/NEXT
M	nombre de chiffres à tirer
N()	liste des chiffres proposés
R$	réponse du joueur



MINI-ANNUAIRE TÉLÉPHONIQUE

Un agenda, c'est bien pratique. Cela peut être aussi très amusant si, en plus des données habituelles (nom, adresse, numéros de téléphone), on met en mémoire des détails plus fantaisistes, tels que date de naissance, taille, pointure des chaussettes, couleur des yeux...

par Jacques Deconchat

Très simplifié, ce programme n'est pas pour autant simpliste. Tel quel, il vous donne la possibilité de trouver le numéro de téléphone à partir d'un nom ou un nom à partir d'un numéro de téléphone ; deux personnes, ou plus, portant le même patronyme seront automatiquement trouvées et affichées.

Il sera, en outre, assez facile de l'étendre, en conservant le principe de départ, pour ajouter des informations sur les personnes répertoriées (adresse, ville, département, âge, etc.) et d'écrire la partie de programme permettant une recherche à partir des nouvelles informations introduites.

```
10  REM  MINI ANNUAIRE TELEPHONIQUE
20  REM  -------------------------
30  REM  LISTE DE VOS CONNAISSANCES
40  REM  -------------------------
50  DATA  LOUIS,16 34 22 61 08,JULES,
    16 61 33 56 72,ETIENNE,16 1 325 36
    35,SOPHIE,16 1 541 22 09
    ,FREDERIQUF.16 1 327 45 21
60  DATA  OCTAVE,16 55 13 22 76,ESTHER,
    16 33 22 10 12,RAYMOND,16 1 232 34
    34,AMELIE,16 76 02 03 04,
    STEPHANE,16 2 345 56 28
```

Ces lignes (en particulier les lignes 50 et 60) contiennent les informations présentes dans le fichier, limité dans cet exemple à dix personnes et deux informations pour chacune (nom et numéro de téléphone). Il est possible d'introduire davantage de renseignements dans chaque DATA (en prenant garde à ne pas faire des lignes trop longues) et de mettre d'autres lignes de DATA : 70, 80... Si l'on a peur de manquer de place, il est bon de savoir que les lignes DATA peuvent être inscrites aussi bien en fin de programme qu'en début, et même éventuellement les deux à la fois.

```
100  REM  DEBUT DU MENU
110  REM  -------------
120  PRINT
130  PRINT "RECHERCHE: 1 PAR NOM"
140  PRINT "           2 PAR TELEPHONE"
150  PRINT "           3 FIN"
160  INPUT R
170  IF R < > 1 AND R < > 2 AND
     R < > 3 THEN 120
180  PRINT
190  ON R GOTO 200,300,400
```

Cette partie est en général désignée en informatique sous le vocable de « menu ». C'est dans ces quelques lignes d'affichage que sont proposées les principales possibilités d'un programme, souvent (comme ici) numérotées, ce qui permet de n'avoir qu'un chiffre à taper pour accéder à telle ou telle fonction (les risques de faute sont ainsi diminués). Dans notre exemple, l'option choisie est enregistrée dans une variable R et contrôlée en ligne 170 (on ne peut taper que 1, 2, 3).

Selon le choix, on sera dirigé sur la ligne 200, la ligne 300 ou la ligne 400.

```
200  REM  RECHERCHE PAR NOM
210  REM  -----------------
220  PRINT "TAPER LE NOM"
230  INPUT N$
240  FOR I = 1 TO 10
250  READ A$,B$
260  IF A$ = N$ THEN  PRINT A$,B$
270  NEXT I
280  RESTORE
290  GOTO 120
```

Première option : rechercher un numéro de téléphone à partir d'un nom. Celui-ci est introduit dans une variable N$. Le fichier tout entier est alors exploré et un test est effectué sur chaque nom (ligne 260) pour savoir s'il correspond au patronyme concerné. Si c'est le cas, on affichera simultanément le nom et le numéro de téléphone.

En fin de lecture, le pointeur de DATA est remis à 0 (instruction RESTORE), cela afin que la prochaine exploration commence de nouveau en début de fichier.

S'il y a plus de dix personnes dans le fichier, il faudra modifier les boucles des lignes 240 et 340, remplacer 10 par le nombre de personnes désirées.

```
300  REM  RECHERCHE PAR TELEPHONE
310  REM  -----------------------
320  PRINT "NUMERO CHERCHE"
330  INPUT N$
340  FOR I = 1 TO 10
350  READ A$,B$
360  IF B$ = N$ THEN  PRINT A$,B$
370  NEXT I
380  GOTO 280
```

La recherche part ici du numéro de téléphone (qui devra être entré exactement de la même façon que dans le fichier).

La fin du programme est réalisée en ligne 400, par une instruction END ; on y accède en tapant 3 en réponse à la question RECHERCHE.

Liste des variables

R	variable de recherche
I	variable de boucle
A$	nom
B$	téléphone
N$	variable utilisée pour la recherche

MINI-ANNUAIRE TÉLÉPHONIQUE - MINI-AN

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
 10  REM  MINI ANNUAIRE TELEPHONIQUE
 20  REM  ---------------------------
 30  REM  LISTE DE VOS CONNAISSANCES
 40  REM  ---------------------------
 50  DATA  LOUIS,16 34 22 61 08,JULES,
     16 61 33 56 72,ETIENNE,16 1 325 36 35,
     SOPHIE,16 1 541 22 09  ,
     FREDERIQUE,16 1 327 45 21
 60  DATA  OCTAVE,16 55 13 22 76,ESTHER,
     16 33 22 10 12,RAYMOND,16 1 232 34 34,
     AMELIE,16 76 02 03 04,
     STEPHANE,16 2 345 56 28
100  REM  DEBUT DU MENU
110  REM  -------------
120  PRINT
130  PRINT "RECHERCHE: 1 PAR NOM"
140  PRINT "           2 PAR TELEPHONE"
150  PRINT "           3 FIN"
160  INPUT R
170  IF R <  > 1 AND R <  > 2 AND
     R <  > 3 THEN 120
180  PRINT
190  ON R GOTO 200,300,400
200  REM  RECHERCHE PAR NOM
210  REM  -----------------
220  PRINT "TAPER LE NOM"
230  INPUT N$
240  FOR I = 1 TO 10
250  READ A$,B$
260  IF A$ = N$ THEN  PRINT A$,B$
270  NEXT I
280  RESTORE
290  GOTO 120
300  REM  RECHERCHE PAR TELEPHONE
310  REM  -----------------------
320  PRINT "NUMERO CHERCHE"
330  INPUT N$
340  FOR I = 1 TO 10
350  READ A$,B$
360  IF B$ = N$ THEN  PRINT A$,B$
370  NEXT I
380  GOTO 280
400  END
```



JONGLEZ AVEC LES MONNAIES

Combien de sucres, de colons, de gourdes ou de roupies pour mon petit billet de cent francs ? Avant de partir pour l'Equateur, le Costa Rica, Haïti ou l'Inde, avec escale à Sanhsonay (les célèbres roupies de Sanhsonay, bien sûr !), devisons gaiement au clavier de notre ordinateur.

par Jacques Deconchat

Il vous est certainement déjà arrivé, lors de voyages à l'étranger, de vous trouver confronté à de redoutables problèmes de change : à l'aller, on y arrive sans trop de mal, mais au retour...

Ce programme vous permettra de surmonter ces difficultés quelles que soient les monnaies concernées, et cela dans tous les sens. Il sera bien sûr plus attrayant de l'utiliser sur un portable, pour vous en servir en temps réel, lorsque vous salivez devant un magnifique manteau de cuir italien ou le dernier gadget électronique japonais. Le prix en est-il vraiment intéressant ? Votre budget supportera-t-il un tel achat ?

Le programme s'articule autour de trois parties. On commence par ce que l'on pourrait appeler « l'initialisation ». L'ordinateur affiche les questions indispensables pour établir la parité monétaire entre les diverses devises.

```
10 REM  CHANGE
20 REM  ------
30 PRINT "MONNAIE A CHANGER"
40 INPUT A$
50 PRINT "POUR QUELLE MONNAIE"
60 INPUT B$
70 PRINT "COMBIEN DE ";A$;" POUR UN ";B$
80 INPUT C
```

On entre le nom de l'une des deux monnaies, le nom de l'autre et enfin la parité (sous la forme : combien de francs faut-il pour obtenir un dollar, par exemple). Les noms des deux monnaies sont enregistrés dans deux variables, A$ et B$, et la parité est conservée dans une troisième variable, C.

On entre ensuite dans la seconde partie du programme (« menu »), qui propose les trois options : change dans un sens, change dans l'autre, ou fin. On répondra en tapant le chiffre correspondant à l'option choisie, puis l'on introduira la somme à changer, qui sera enregistrée dans une variable S.

```
90  PRINT "CHANGE : 1(DE ";A$;
    " VERS ";B$;"); 2(DE ";B$;
    " VERS ";A$;"); 3(FIN)"
100  INPUT R
110  IF R = 3 THEN  END
120  PRINT "SOMME A CHANGER"
130  INPUT S
140  ON R GOTO 200,300
```

La seule difficulté réside dans l'aiguillage de la ligne 140, qui dirige sur deux programmes différents selon la valeur introduite dans la variable R de la ligne 100. Sur certains appareils (ZX 81 ou Spectrum, en particulier), la ligne 140 devra être remplacée par
140 GOTO 100+R 100.

```
200 REM  CHANGE 1
210 REM  --------
220 LET T =  INT (100 * S / C) / 100
230 PRINT "POUR ";S;" ";A$;
    " VOUS AUREZ ";T;" ";B$
    240  GOTO 90
```

Le calcul du résultat du change est fait en ligne 220, par une formule qui permet d'obtenir un affichage n'ayant pas plus de deux chiffres derrière la virgule (on prend le résultat exact, on le multiplie par 100 ; on prend alors sa partie entière, ce qui élimine tous les chiffres qui pourraient se trouver derrière la virgule, et on redivise par 100 pour retrouver le résultat initial, avec au plus deux chiffres derrière la virgule).

Le résultat du calcul est présenté par une phrase, puis on repart pour une nouvelle conversion (on termine en tapant 3).

```
300 REM  CHANGE 2
310 REM  --------
320 LET T =  INT (100 * S * C) / 100
330 PRINT "POUR ";S;" ";B$;
    " VOUS AUREZ ";T;" ";A$
340 GOTO 90
```

Cette partie, identique à la précédente, permet d'effectuer le change dans l'autre sens.

```
MONNAIE A CHANGER
?FRANCS
POUR QUELLE MONNAIE
?DOLLARS
COMBIEN DE FRANCS POUR UN DOLLAR
?8.5
CHANGE : 1(DE FRANCS VERS DOLLARS); 2(DE DOLLARS VERS FRANCS); 3(FIN)
?2
SOMME A CHANGER
?100
POUR 100 DOLLARS VOUS AUREZ 850 FRANCS
CHANGE : 1(DE FRANCS VERS DOLLARS); 2(DE DOLLARS VERS FRANCS); 3(FIN)
?1
SOMME A CHANGER
?10000
POUR 10000 FRANCS VOUS AUREZ 1176.47 DOLLARS
CHANGE : 1(DE FRANCS VERS DOLLARS); 2(DE DOLLARS VERS FRANCS); 3(FIN)
?3
```

FICHE PROGRAMME

JONGLEZ AVEC LES MONNAIES - JONGLEZ

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10  REM  CHANGE
20  REM  ------
30  PRINT "MONNAIE A CHANGER"
40  INPUT A$
50  PRINT "POUR QUELLE MONNAIE"
60  INPUT B$
70  PRINT "COMBIEN DE ";
    A$;" POUR UN ";B$
80  INPUT C
90  PRINT "CHANGE :
    1(DE ";A$;" VERS ";B$;"),
    2(DE ";B$;" VERS ";A$;");
    3(FIN)"
100  INPUT R
110  IF R = 3 THEN  END
120  PRINT "SOMME A CHANGER"
130  INPUT S
140  ON R GOTO 200,300
200  REM  CHANGE 1
210  REM  --------
220  LET T =
     INT (100 * S / C) / 100
230  PRINT "POUR ";S;" ";A$;
     " VOUS AUREZ ";T;" ";B$
240  GOTO 90
300  REM  CHANGE 2
310  REM  --------
320  LET T =  INT
     (100 * S * C) / 100
330  PRINT "POUR ";S;" ";
     B$;" VOUS AUREZ ";T;" ";A$
340  GOTO 90
```

Liste des variables

A$	première devise
B$	deuxième devise
C	taux de change
R	choix de l'option
S	somme initiale
T	somme après le change



LE DEVIN

Plus de gui ni de serpes d'or pour les Panoramix des temps informatiques ! Votre Ordinateur propose plutôt un bon petit programme pour jouer au devin. Et tant mieux pour vous si, en plus, vous êtes télépathe !

par Jacques Deconchat

Vous êtes enfermé dans une pièce complètement noire. Dans la pièce voisine, votre partenaire se concentre sur un nombre qu'il essaie de vous transmettre par la seule puissance de son cerveau. Soudain, vous sortez et annoncez aux spectateurs éberlués LE nombre ! Qui de nous n'a pas un jour voulu vérifier ses dons de transmission de pensée ? Mais faut-il réellement qu'il y ait pensée à transmettre ? Ne s'agit-il pas plutôt d'une intuition géniale ? Pour le savoir, vous pouvez faire appel à votre ordinateur.

Ce petit programme va tirer vingt fois le chiffre 0 ou 1, le premier associé symboliquement au côté face d'une pièce, le second au côté pile. Vous devez essayer de deviner la suite ainsi obtenue. Si vous recevez régulièrement la mention « brillant », alors il faudra réellement vous poser des questions sur vos capacités divinatoires !

On commence par quelques phrases d'explication concernant les règles du jeu proposé.

```
30  PRINT "CE PROGRAMME VA
    LANCER 20 PIECES DE MONNAIE"
40  PRINT "APRES CHAQUE LANCER ,
    IL VOUS PROPOSE DE DIRE"
50  PRINT "SI LA PIECE EST TOMBEE
    SUR PILE (P) OU FACE (F)"
60  PRINT "VOTRE CAPACITE DE DEVIN
    SERA AFFICHEE EN FIN DE PARTIE"
70  PRINT "ETES-VOUS PRET ( O/N) ?"
80  INPUT R$
90  IF R$ <  > "O" THEN  RUN
```

La ligne 70 utilise un « truc » pour vous laisser le temps de lire les lignes précédentes. Il suffira de taper 0 en réponse à la question « ETES-VOUS PRET » pour que le jeu commence.

```
100  REM  DEBUT DES LANCERS
110  LET S = 0
120  FOR I = 1 TO 20
130  LET H =  INT ( RND (1) * 2)
140  PRINT "LANCER NO.";I;
     " - P(ILE) OU F(ACE) ?"
150  INPUT R$
160  IF R$ <  > "P" AND
     R$ <  > "F" THEN  GOTO 140
170  IF R$ = "P" THEN  LET E = 1
180  IF R$ = "F" THEN  LET E = 0
190  IF E = H THEN  LET S = S + 1
200  NEXT I
```

La variable S, mise à 0 en ligne 100, servira à tenir le compte des résultats exacts.

La boucle I (lignes 120 à 200) assure la série de vingt lancers.

Le tirage de la pièce par l'ordinateur est fait par la ligne 130, à l'aide d'une variable H qui peut prendre les valeurs 0 ou 1. Cette ligne devra être modifiée sur certains appareils selon le type de variable aléatoire définie par le constructeur.

La ligne 140 affiche le numéro du lancer et demande au joueur de préciser sa réponse par un P ou un F qui sera saisi dans une variable R$ puis transformé en nombre dans une variable E (1 ou 0) par les lignes 170 et 180. Le résultat est comparé avec le tirage de l'ordinateur et le score est établi en conséquence (ligne 190).

```
210  LET F =  INT ( SQR (S))
220  PRINT
230  PRINT "VOUS ETES UN DEVIN ";
240  ON F GOTO 300,320,340,360
300  PRINT "NUL"
310  END
320  PRINT "MEDIOCRE"
330  END
340  PRINT "MOYEN"
350  END
360  PRINT "BRILLANT"
370  END
```

La ligne 210 permet, en prenant la partie entière de la racine carrée de S, d'obtenir un nombre pouvant aller de 0 à 4 (0 est très improbable, une seule réponse exacte suffisant pour avoir 1). En fonction de ce nombre, un message sera affiché, qui tiendra compte du total des bonnes réponses que vous aurez données (il faut au moins seize réponses correctes pour se voir attribuer la mention « BRILLANT »). Sur le ZX 81 et le Spectrum, la ligne 240 doit être remplacée par 240 GOTO 300+F 20.

```
CE PROGRAMME VA LANCER
20 PIECES DE MONNAIE
APRES CHAQUE LANCER, IL
VOUS PROPOSE DE DIRE SI LA
PIECE EST TOMBEE SUR PILE
(P) OU FACE (F)
VOTRE CAPACITE DE DEVIN
SERA AFFICHEE EN FIN DE
PARTIE
ETES-VOUS PRET (O/N)?
?0
LANCER NO.1 – P(ILE) OU
F(ACE)?
```

Liste des variables

- R$ réponse OUI ou NON
- S score
- I variable de boucle
- H tirage de l'ordinateur (0 ou 1)
- E prévision du joueur
- F variable utilisée pour l'affichage des messages de fin

LE DEVIN-LE DEVIN-LE DEVIN-LE D

Attention ! Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

```
10  REM  SERIEZ-VOUS UN BON DEVIN
20  REM  -------------------------
30  PRINT "CE PROGRAMME VA LANCER
    20 PIECES DE MONNAIE"
40  PRINT "APRES CHAQUE LANCER ,
    IL VOUS PROPOSE DE DIRE"
50  PRINT "SI LA PIECE EST TOMBEE
    SUR PILE (P) OU FACE (F)"
60  PRINT "VOTRE CAPACITE DE DEVIN
    SERA AFFICHEE EN FIN DE PARTIE"
70  PRINT "ETES-VOUS PRET ( O/N) ?"
80  INPUT R$
90  IF R$ <  > "O" THEN  RUN
100  REM  DEBUT DES LANCERS
110  LET S = 0
120  FOR I = 1 TO 20
130  LET H =  INT ( RND (1) * 2)
140  PRINT "LANCER NO.";I;"
     - P(ILE) OU F(ACE) ?"
150  INPUT R$
160  IF R$ <  > "P" AND R$ <  >
     "F" THEN  GOTO 140
170  IF R$ = "P" THEN  LET E = 1
180  IF R$ = "F" THEN  LET E = 0
190  IF E = H THEN  LET S = S + 1
200  NEXT I
210  LET F =  INT ( SQR (S))
220  PRINT
230  PRINT "VOUS ETES UN DEVIN ";
240  ON F GOTO 300,320,340,360
300  PRINT "NUL"
310  END
320  PRINT "MEDIOCRE"
330  END
340  PRINT "MOYEN"
350  END
360  PRINT "BRILLANT"
370  END
```



L'ESPIONNITE

Du nouveau dans les fiches programmes de *V.O. L'espionnite* est entièrement réalisée à partir du Logo. Ce langage informatique, très apprécié des pédagogues, a recours aux services de la célèbre tortue. Logo a été développé par Seymour Papert, informaticien en renom, d'après les théories du psychologue suisse Jean Piaget. Les non-initiés consulteront avec profit la rubrique permanente de *Votre Ordinateur* qui présente ce langage.

par Maxime Meystre

Lorsqu'un ordinateur échange des informations avec un autre ordinateur, il émet une série de « bruits » caractéristiques que tout espion peut capter pour percer le secret des messages. Un seul recours : ne pas les confier aux machines sans les coder, sous peine de voir le produit de votre dur labeur d'apprenti programmeur passer aux mains d'indélicats pirates des temps modernes.

En Logo, une phrase est une liste de mots (voir leçon de Logo p. 68). Nous nommerons p notre phrase, m un mot, c un caractère.

Problème n° 1 : coder une phrase.

Une phrase tapée normalement apparaîtra en code et inversement. Ainsi, l'expression PAR EXEMPLE donnera SEWLFNWAXR, après avoir été codée.

```
pour coder.phrase      :p
si      :p = [ ] [retourne[ ] ]
retourne phrase
        coder.mot premier    :p coder.phrase sp    :p
fin
```

Coder une phrase consiste à reconstituer une phrase à partir du codage de ses mots.
On prend donc le premier mot de p – premier :p –. On le code et on recommence le même processus avec le premier mot de p lorsque l'élément codé a déjà été enlevé – coder.phrase saufpremier (sp) :p –, et ce jusqu'à ce qu'il n'y ait plus de mots à coder, c'est-à-dire lorsque p est vide – si :p = [ ]... –

Problème n° 2 : coder un mot.
Le codage d'un mot utilise le raisonnement analysé dans le problème n° 1

```
pour coder .mot       :m
si    :m = " [retourne"]
retourne mot
        coder.cara premier :m    coder.mot sp :m
fin
```

Problème n° 3 : coder un caractère.
Pour coder un caractère, il suffit de prendre son code et de lui ajouter une constante k, puis de le retourner sous forme de caractère.

```
pour coder.cara       :c        :k
donne "cl      ascii   :c       +      :k
si   :cl      >      91      [donne "cl        :cl – 26]
retourne car    :cl
fin
```

Il ne reste plus qu'à définir une loi pour k et à l'introduire dans nos trois procédures, au moyen d'un test de fin de codage.

```
pour coder
donne "l lisliste
si   :l = [FIN] [stop]
affiche coder.phrase   :l
coder
fin
pour coder.phrase   :p
si   :p = [ ] [retourne[ ] ]
retourne phrase    coder.mot premier :p
```

SUITE →

ESPIONNITE - L'ESPIONNITE - L'ESPIO

Attention ! Ce programme en Logo est conçu pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente peut-être des particularités : dans ce cas, nous vous demandons de nous les signaler.

Suite

```
          compte premier :p   coder.phrase   sp :p
fin
pour coder.mot   :m   :k
si  :m   =   "[retourne"]
retourne mot coder.cara premier   :m   :k
coder.mot   sp   :m   :k + 1
fin
pour coder.cara   :c   :k
donne "cl ascii   :c +   :k
si   :cl >91
[retourne car (   :cl – 26)]
[retourne car   :cl]
fin
```

Nous avons introduit la constante k comme paramètre de coder.mot et nous lui avons donné comme valeur intitiale compte premier :p. Ainsi pour coder le mot AVION, k aura la valeur 5, nombre de lettres d'avion. k évolue ensuite en ajoutant 1 à chaque caractère.
AVION : A décalé de 5, V décalé de 6... N décalé de 8

Le décodage relève de la même logique : simplement, on retranchera k au lieu de l'ajouter, et on ajoutera au résultat 26 si le nombre obtenu est inférieur à 65 – code de A.

```
pour décoder.cara   :c   :k
donne "cl ascii   :c –   :k
si   :cl   < 65
[retourne car (:cl + 26)
[retourne car   :l]
fin
```

Ce qu'il faut savoir

✶ **pour** et **fin** servent à définir le début et la fin d'une procédure.
✶ **donne** crée une variable et lui donne une valeur.
✶ "**k** est la variable k, :k est son contenu.
✶ **retourne** élabore un résultat et le retourne à la procédure appelante.
✶ **lisliste** transforme une phrase tapée au clavier en liste.
✶ **ascii** donne le code du caractère, **car** donne le caractère correspondant au code.
✶ **mot, phrase, premier, saufpremier, compte** : voir la leçon Logo de ce numéro.

Le programme a été écrit pour Dr Logo (Digital Research, actuellement sur IBM-PC). Tout est écrit en minuscules. Tous les autres Logo (sauf Logo EN1) s'écrivent en majuscules. De plus, une ligne qui commence par une espace est la suite de la ligne précédente, ce qui n'est pas vrai dans les autres versions.

VARIANTES

✶ **lisliste → LISLIGNE (Commodore-Q, TI-Logo, Edi-Logo)**
LISLISTE (Apple-Logo)
LL (Thomson)

✶ **si ... [ ] [ ] → si ... alors [ ] ... sinon [ ] (Logo EN1)**
SI ... [ ] [ ] (Apple-Logo, Thomson, Goupil)
SI ... ALORS ... SINON (Commodore-Q, TI-Logo, Edi-Logo)

✶ **retourne → RETOURNE (Commodore-Q, Apple-Logo)**
SORS (TI-Logo, Edi-Logo)
RET (Thomson)
rends (Logo EN1)

Les amateurs de Logo consulteront bénéfiquement le « Dictionnaire Logo » de Gérard Bossuet publié aux éditions Sybex, 250 FF environ.



LE TAQUIN

Vous vous rappelez le « pousse-pousse » sur lequel vous vous crispiez les doigts, profitant du vide d'une case pour replacer les autres dans le bon ordre ?

par Jean-François Sehan

Le taquin ou puzzle à 15 a été inventé par Sam Lloyd, aux États-Unis, dans les années 1880. Il se compose d'un plateau et de quinze plaquettes numérotées de 1 à 15 que l'on dispose en désordre (plus de vingt mille milliards de possibilités). Une case vide sur le plateau permet de bouger les plaquettes pour remettre les chiffres dans un ordre croissant. Seules les plaquettes voisines de celle-ci peuvent se déplacer.

Il ne s'agit pas ici du jeu original de Sam Lloyd. En conséquence, toutes les positions de départ permettent une reconstitution de l'ensemble du jeu.

Le tableau T correspond aux seize cases du taquin. Après son dimensionnement, on y place les numéros de 1 à 15 (lignes 100 à 130). Le tableau D correspond aux déplacements des plaquettes (moins une case pour la gauche, plus quatre cases pour le bas, etc.).

```
10 REM TAQUIN
20 REM -------------------
30 DIM T(16)
40 DIM D(4)
50 LET D(1)=-4
60 LET D(2)=4
70 LET D(3)=1
80 LET D(4)=-1
90 REM -------------------
100 REM MELANGE DU TAQUIN
110 FOR I=1 TO 15
120 LET T(I)=I
130 NEXT I
```

On mélange les numéros en permutant deux cases désignées par un nombre aléatoire et le pointeur de la case vide (variable V). Cent vingt-huit permutations sont nécessaires pour mélanger correctement le taquin.

```
140 LET V=16
150 FOR I=1 TO 128
160 LET A=INT(RND(1)*4)+1
170 IF V+D(A)<1 OR V+D(A)>16 THEN 160
180 IF INT(V/4)=V/4 AND A=3 THEN 160
190 IF INT(V/4)=(V-1)/4 AND A=4 THEN 160
200 LET T(V)=T(V+D(A))
210 LET T(V+D(A))=0
220 LET V=V+D(A)
230 NEXT I
```

On affiche le jeu à l'écran dans le nouvel ordre – ou plutôt désordre !

```
240 REM ----------------------------
250 REM DESSIN DU TAQUIN
260 PRINT:PRINT
270 PRINT "+----+----+----+----+"
280 FOR I=1 TO 13 STEP 4
290 FOR J=0 TO 3
300 IF T(I+J)=0 THEN 330
310 PRINT TAB((J*5)+1);"I";T(I+J);
320 GOTO 340
330 PRINT TAB((J*5)+1);"I";
340 NEXT J
350 PRINT TAB(21);"I"
360 PRINT "+----+----+----+----+"
370 NEXT I
```

On déplace le numéro tapé par le joueur en vérifiant qu'il n'est pas inférieur à 1 ou supérieur à 15, et qu'il est bien placé à côté de la case vide. Le déplacement est calculé avec le tableau D.

```
380 REM -----------------------
390 REM DEPLACEMENT
400 PRINT "QUEL NUMERO ?"
410 INPUT R
420 IF R<1 OR R>15 THEN 400
430 FOR I=1 TO 16
440 IF R=T(I) THEN 460
450 NEXT I
460 LET R=I
470 FOR I=1 TO 4
480 IF R+D(I)>16 OR R+D(I)<1 THEN
                                500
490 IF T(R+D(I))=0 THEN 520
500 NEXT I
510 GOTO 400
520 LET T(R+D(I))=T(R)
530 LET T(R)=0
```

A chaque essai, on vérifie qu'il reste des plaquettes dans le désordre. Si c'est le cas, on retourne à la ligne 260 pour un nouvel essai.

```
540 REM -----------------------
550 REM FIN DE PARTIE ?
560 FOR I=1 TO 15
570 IF T(I)<>I THEN 260
580 NEXT I
590 END
```

2	6	8	3
1	4	7	
12	5	14	15
9	13	10	11

QUEL NUMÉRO ?

? 7

LE TAQUIN - LE TAQUIN - LE TA

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler

```
10 REM TAQUIN
20 REM -----------------------------
30 DIM T(16)
40 DIM D(4)
50 LET D(1)=-4
60 LET D(2)=4
70 LET D(3)=1
80 LET D(4)=-1
90 REM -----------------------------
100 REM MELANGE DU TAQUIN
110 FOR I=1 TO 15
120 LET T(I)=I
130 NEXT I
140 LET V=16
150 FOR I=1 TO 128
160 LET A=INT(RND(1)*4)+1
170 IF V+D(A)<1 OR V+D(A)>16 THEN 160
180 IF INT(V/4)=V/4 AND A=3 THEN 160
190 IF INT(V/4)=(V-1)/4 AND A=4 THEN 160
200 LET T(V)=T(V+D(A))
210 LET T(V+D(A))=0
220 LET V=V+D(A)
230 NEXT I
240 REM ---------------------------
250 REM DESSIN DU TAQUIN
260 PRINT:PRINT
270 PRINT "+----+----+----+----+"
280 FOR I=1 TO 13 STEP 4
290 FOR J=0 TO 3
300 IF T(I+J)=0 THEN 330
310 PRINT TAB((J*5)+1);"I";T(I+J);
320 GOTO 340
330 PRINT TAB((J*5)+1);"I";
340 NEXT J
350 PRINT TAB(21);"I"
360 PRINT "+----+----+----+----+"
370 NEXT I
380 REM ---------------------------
390 REM DEPLACEMENT
400 PRINT "QUEL NUMERO ?"
410 INPUT R
420 IF R<1 OR R>15 THEN 400
430 FOR I=1 TO 16
440 IF R=T(I) THEN 460
450 NEXT I
460 LET R=I
470 FOR I=1 TO 4
480 IF R+D(I)>16 OR R+D(I)<1 THEN 500
490 IF T(R+D(I))=0 THEN 520
500 NEXT I
510 GOTO 400
520 LET T(R+D(I))=T(R)
530 LET T(R)=0
540 REM ---------------------------
550 REM FIN DE PARTIE ?
560 FOR I=1 TO 15
570 IF T(I)<>I THEN 260
580 NEXT I
590 END
```

Liste des variables

R	Choix du joueur
T()	Tableau des emplacements des plaquettes
V	Pointe la case vide sur le plateau
A	Nombre aléatoire pour le mélange
D()	Tableau des déplacements des plaquettes
I	Indice de boucle FOR/NEXT

LE SOUS-MARIN

Un sous-marin de nationalité inconnue s'est introduit dans les eaux territoriales. Partez vite à sa recherche.

par Jean-François Sehan

La partie se déroule sur une grille de 20 cases sur 20. Vous avez toute liberté dans vos déplacements (toutes directions et toutes distances). A chaque essai, la distance entre votre bateau et le sous-marin s'affiche, ainsi que vos coordonnées du moment. La distance étant relative, il est possible qu'elle soit négative (c'est toujours le cas en début de partie).

Sentant le danger, le sous-marin se déplace d'une case, dans n'importe quelle direction, à chacun de vos mouvements. Et dès qu'il est repéré, le programme affiche le message : « sous-marin à la verticale ». Il suffit alors de rentrer au clavier la profondeur d'explosion de la grenade (un chiffre entre 1 et 3). Si ce chiffre est incorrect, le submersible pourchassé se déplace d'une case et le jeu reprend normalement. Si le sous-marin est touché, le programme affiche le message : « sous-marin coulé ». Mais la bataille continue inexorablement, car un autre intrus arrive à la rescousse.

Les variables X et Y correspondent aux coordonnées du bateau. On place dans les variables V et W (coordonnées du sous-marin) des nombres aléatoires compris entre 3 et 19 pour que les deux bâtiments soient assez distants l'un de l'autre.

```
10 REM SOUS-MARIN
20 REM --------------------
30 LET X=1
40 LET Y=1
50 LET V=INT(RND(1)*17)+3
60 LET W=INT(RND(1)*17)+3
```

Les lignes 90 à 140 incrémentent les coordonnées du sous-marin en ajoutant à celles-ci un nombre aléatoire compris entre −1 et 1, tout en vérifiant que l'on ne sort pas de la grille du jeu.

```
70 REM --------------------------
80 REM DEPLACEMENT S.-MARIN
90 LET A=INT(RND(1)*3)-1
100 IF V+A<1 OR V+A>20 THEN 90
110 LET V=V+A
120 LET A=INT(RND(1)*3)-1
130 IF W+A<1 OR W+A>20 THEN 120
140 LET W=W+A
```

Les lignes 170 à 200, elles, calculent la distance entre le bateau et le sous-marin. On prend comme distance la différence la plus importante entre les coordonnées. Cette distance (variable D) est affichée à la ligne 220.

```
150 REM --------------------------
160 REM DISTANCE
170 IF ABS(X-V)>ABS(Y-W) THEN 200
180 LET D=Y-W
190 GOTO 220
200 LET D=X-V
210 PRINT
220 PRINT "DISTANCE:";D
```

Après affichage des anciennes coordonnées, on saisit au clavier les nouvelles valeurs en vérifiant, comme pour le sous-marin, que l'on ne sort pas du jeu. Si les coordonnées du bateau et du sous-marin sont différentes, on retourne au déplacement du sous-marin (ligne 90).

```
230 REM --------------------------
240 REM DEPLACEMENT BATEAU
250 PRINT "COORDONNEES BATEAU:"
260 PRINT "X=";X;"    Y=";Y
270 PRINT "NOUVELLE VALEUR DE X ?"
280 INPUT X
290 IF X<1 OR X>20 THEN 270
300 PRINT "NOUVELLE VALEUR DE Y ?"
310 INPUT Y
320 IF Y<1 OR Y>20 THEN 300
330 IF X<>V OR Y<>W THEN 90
```

```
DISTANCE : −14
COORDONNÉES BATEAU :
X = 1     Y = 1
NOUVELLE VALEUR DE X ?
? 10
NOUVELLE VALEUR DE Y ?
? 10
```

Lorsqu'elles sont les mêmes, on saisit au clavier la profondeur d'explosion des grenades. Si cette valeur correspond à la profondeur du sous-marin (variable S), on affiche le message « sous-marin coulé » avant de revenir au début du programme (ligne 30). Dans le cas contraire, on continue le jeu normalement.

```
340 REM --------------------------
350 REM PROFONDEUR
360 LET S=INT(RND(1)*3)+1
370 PRINT "SOUS-MARIN A LA VERTICALE"
380 PRINT "PROFONDEUR ?"
390 INPUT P
400 IF P<1 OR P>3 THEN 380
410 IF P<>S THEN 100
420 PRINT "SOUS-MARIN COULE"
430 PRINT "*****************"
440 GOTO 30
```

Liste des variables

A	Nombre aléatoire pour le déplacement du sous-marin
D	Distance bateau/sous-marin
P	Profondeur choisie par le joueur
S	Profondeur du sous-marin
V et W	Coordonnées du sous-marin
X et Y	Coordonnées du bateau

LE SOUS-MARIN - LE SOUS-MA

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler

```
10 REM SOUS-MARIN
20 REM ----------------------------
30 LET X=1
40 LET Y=1
50 LET V=INT(RND(1)*17)+3
60 LET W=INT(RND(1)*17)+3
70 REM ----------------------------
80 REM DEPLACEMENT S.-MARIN
90 LET A=INT(RND(1)*3)-1
100 IF V+A<1 OR V+A>20 THEN 90
110 LET V=V+A
120 LET A=INT(RND(1)*3)-1
130 IF W+A<1 OR W+A>20 THEN 120
140 LET W=W+A
150 REM ---------------------------
160 REM DISTANCE
170 IF ABS(X-V)>ABS(Y-W) THEN 200
180 LET D=Y-W
190 GOTO 220
200 LET D=X-V
210 PRINT
220 PRINT "DISTANCE:";D
230 REM ---------------------------
240 REM DEPLACEMENT BATEAU
250 PRINT "COORDONNEES BATEAU:"
260 PRINT "X=";X;"   Y=";Y
270 PRINT "NOUVELLE VALEUR DE X ?"
280 INPUT X
290 IF X<1 OR X>20 THEN 270
300 PRINT "NOUVELLE VALEUR DE Y ?"
310 INPUT Y
320 IF Y<1 OR Y>20 THEN 300
330 IF X<>V OR Y<>W THEN 90
340 REM ---------------------------
350 REM PROFONDEUR
360 LET S=INT(RND(1)*3)+1
370 PRINT "SOUS-MARIN A LA VERTICALE"
380 PRINT "PROFONDEUR ?"
390 INPUT P
400 IF P<1 OR P>3 THEN 380
410 IF P<>S THEN 100
420 PRINT "SOUS-MARIN COULE"
430 PRINT "****************"
440 GOTO 30
```

LE P'TIT BAC

Ce jeu de société classique n'a pas fini de déployer ses charmes à qui veut tromper l'attente. Ce programme propose de l'agrémenter d'une trouvaille qui permettra de jouer au « cadavre exquis » du mouvement surréaliste.

par Jacques Deconchat

La règle est assez élémentaire, du moins dans la version qui est proposée ici.
L'ordinateur affiche une lettre, prise dans une liste (non limitative) établie en début de programme.

```
30  LET A$ = "ABCDEILMNOPR"
40  LET X =  INT ( RND (1) * 12 + 1)
50 L$ =  MID$ (A$,X,1)
60  PRINT "LA LETTRE TIREE EST : ";L$
```

La ligne 30 propose une liste de lettres. La ligne 40 assure le tirage au sort de l'une de ces lettres. Si l'on désire ajouter ou ôter des lettres, il faudra changer la variable A$ et modifier en ligne 40 le nombre de lettres inscrites (ici 12).
Sur certains appareils, le tirage au sort devra être fait avec RND ou encore RND(O). De même, la ligne 50 (L$=MID$(A$,X,1) devra être remplacée par L$=A$(X) (sur ZX 81 ou Spectrum) ou par L$=SEG(A$,X,1) sur TI 99.
Après l'affichage de la lettre tirée, le programme pose trois questions : un prénom de fille, un prénom de garçon, un nom de ville. Bien entendu, les réponses devront commencer par la lettre sortie (c'est ce qui fait l'intérêt du jeu). On peut augmenter la difficulté en posant davantage de questions et en laissant à chacun des joueurs un temps limité pour répondre.

```
70  PRINT "DONNEZ UN PRENOM DE FILLE"
80  INPUT P$
90  IF  LEFT$ (P$,1) <  > L$ THEN  GOTO 70
100  PRINT "DONNEZ UN PRENOM DE GARCON"
110  INPUT G$
120  IF  LEFT$ (G$,1) <  > L$ THEN  GOTO 100
130  PRINT "DONNEZ UNE VILLE"
140  INPUT V$
150  IF  LEFT$ (V$,1) <  > L$ THEN  GOTO 130
```

Les trois lignes 90, 120 et 150 permettent de s'assurer que la réponse proposée par le joueur commence bien par la lettre indiquée ; si ce n'est pas le cas, la question sera posée de nouveau.
Sur les ZX 81 et Spectrum, on devra remplacer **LEFT$(X$,1)** par **X$(1)** ; sur TI 99 par **SEG(X$,1,1)**.
Le jeu de société qui a inspiré le programme consiste usuellement à poser les mêmes questions à chacun des participants, jusqu'à ce que l'un d'eux reste coi et soit de ce fait éliminé du jeu.
Il m'a semblé plus intéressant de réaliser une version légèrement différente où l'ordinateur intervient pour afficher une petite phrase fabriquée à partir des réponses et d'un verbe tiré au hasard d'une liste préétablie.
Bien entendu, la liste proposée pourra être modifiée en vue de résultats plus farfelus ou plus amusants.

```
160 Z =  INT ( RND (1) * 10 + 1)
170  FOR I = 1 TO Z
180  READ U$
190  NEXT I
200  PRINT
210  PRINT P$;" ET ";G$;" ";U$;" A ";V$
220  RESTORE
230  GOTO 10
240  DATA   HABITENT,MANGENT,DORMENT,SONT NES,
     NE REVIENDRONT JAMAIS,S'AIMENT,ONT OUBLIE
     LEURS PORTEFEUILLES
250  DATA  VONT EN VOYAGE,SE SONT RENCONTRES,
     ONT L'HONNEUR DE VOUS FAIRE PART DE LEUR
     MARIAGE
```

Si l'on enlève ou si l'on rajoute des verbes, il sera prudent de changer, en ligne 160, le nombre 10, qui indique le nombre de verbes disponibles.
Le ZX 81 ne dispose pas des trois instructions DATA, READ, et RESTORE. On peut cependant les simuler, dans de nombreux cas, en faisant les modification suivantes :

```
160 Z = INT(RND  10 + 1)
170 B$="HABITENT , MANGENT,..... DE LEUR MARIAGE"
180 LET U$ = ""
190 FOR I = 1 TO LEN(B$)
200 IF B$(I) = "," THEN LET Z = Z - 1
210 IF Z = 1 THEN LET U$ = ""
220 IF Z = 0 THEN GOTO 250
230 LET U$ = U$ + B$(I)
240 NEXT I
250 PRINT P$;" ET ";G$;" ";U$;" A ";V$
260 GOTO 10
```

LE P'TIT BAC - LE P'TIT BAC - LE

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler

```
10  REM  LE JEU DU BACCALAUREAT
20  REM  -----------------------
30  LET A$ = "ABCDEILMNOPR"
40  LET X =  INT ( RND (1) * 12 + 1)
50 L$ =  MID$ (A$,X,1)
60  PRINT "LA LETTRE TIREE EST : ";L$
70  PRINT "DONNEZ UN PRENOM DE FILLE"
80  INPUT P$
90  IF  LEFT$ (P$,1) <  > L$ THEN  GOTO 70
100  PRINT "DONNEZ UN PRENOM DE GARCON"
110  INPUT G$
120  IF  LEFT$ (G$,1) <  > L$ THEN  GOTO 100
130  PRINT "DONNEZ UNE VILLE"
140  INPUT V$
150  IF  LEFT$ (V$,1) <  > L$ THEN  GOTO 130
160 Z =  INT ( RND (1) * 10 + 1)
170  FOR I = 1 TO Z
180  READ U$
190  NEXT I
200  PRINT
210  PRINT P$;" ET ";G$;" ";U$;" A ";V$
220  RESTORE
230  GOTO 10
240  DATA  HABITENT,MANGENT,DORMENT,SONT NES,
     NE REVIENDRONT JAMAIS,S'AIMENT,ONT
     OUBLIE LEURS PORTEFEUILLES
250  DATA  VONT EN VOYAGE,SE SONT RENCONTRES,
     ONT L'HONNEUR DE VOUS FAIRE PART
     DE LEUR MARIAGE
```

Liste des variables

A$	lettres à afficher
X	tirage d'une lettre au hasard
L$	lettre tirée
P$	premier prénom
G$	deuxième prénom
V$	ville
Z	tirage d'un verbe au hasard
I	variable de boucle
U$	verbe

SUPER NOMBRE SECRET

Découvrez le nombre secret proposé par votre ordinateur en effectuant des opérations en cascades.

par Jean-François Sehan

Le nombre choisi est compris entre 1 et 99. A chaque solution erronée proposée, le programme répond : « trop petit » ou « trop grand », vous situant ainsi par rapport au nombre à trouver.

Première difficulté du jeu : toute proposition doit être le résultat d'une opération. Par exemple : 6 × 3 = 18 (nombre que vous proposez).

Pour commencer, vous choisissez deux chiffres différents de 1 à 9, que vous multipliez, divisez, additionnez ou soustrayez l'un de l'autre. Tout au long du jeu, les opérateurs seront les chiffres de 1 à 9. En revanche, vous devez vous servir du résultat de votre première opération pour effectuer la deuxième, puis du résultat de la deuxième pour effectuer la troisième...

Seconde difficulté : il est interdit d'utiliser deux fois le même chiffre comme opérateur. Les lignes 30 à 70 remplissent le tableau A(1) à A(9) avec le nombre 1 pour connaître les chiffres déjà employés. La ligne 80 place dans la variable N le nombre secret (nombre aléatoire). Puis l'on saisit le premier des chiffres et l'on va directement en ligne 460 : ce chiffre est peut-être le nombre à découvrir.

```
10 REM SUPER NOMBRE SECRET
20 REM ---------------------
30 DIM A(9)
40 LET C=0
50 FOR I=1 TO 9
60 LET A(I)=1
70 NEXT I
80 LET N=INT(RND(1)*99)+1
90 PRINT "PREMIER CHIFFRE ?"
100 INPUT S
110 IF S<1 OR S>9 THEN 90
120 LET R=S
130 GOTO 460
```

PREMIER CHIFFRE ?
? 5
NOMBRE PROPOSÉ : 5
C'EST TROP PETIT

Les lignes 160 à 200 affichent le tableau A(1) à A(9) pour les chiffres restants.

```
140 REM --------------
150 REM LE JEU
160 PRINT "IL VOUS RESTE:"
170 FOR I=1 TO 9
180 IF A(I)=1 THEN PRINT I,
190 NEXT I
200 PRINT
```

IL VOUS RESTE :
1 2 3 4 6 7 8 9

Le programme saisit alors le chiffre et l'opération à utiliser. Si vous avez triché, on retourne en ligne 160 pour une nouvelle saisie.

Les lignes 250 à 430 recherchent le signe que vous voulez utiliser et effectuent les calculs.

```
210 PRINT "NOMBRE CHOISI:"
220 INPUT S
230 IF S<1 OR S>9 THEN 210
240 IF A(S)=0 THEN 160
250 PRINT "OPERATION (+,-,*
                     OU /)"
260 INPUT O$
270 IF O$="+" THEN 320
280 IF O$="-" THEN 350
290 IF O$="*" THEN 380
300 IF O$="/" THEN 410
310 GOTO 160
320 IF R+S>99 THEN 160
330 LET R=R+S
340 GOTO 460
350 IF R-S<1 THEN 160
360 LET R=R-S
370 GOTO 460
380 IF R*S>99 THEN 160
390 LET R=R*S
400 GOTO 460
410 IF INT(R/S)<>R/S THEN 160
420 IF R/S<1 THEN 160
430 LET R=R/S
```

NOMBRE CHOISI :
? 9
OPÉRATION (+, –, * OU /)
? *

Les lignes 470 à 480 comparent votre solution au nombre secret et renvoient au message correspondant. Si vous avez eu de la chance, la ligne 580 vous félicitera, sinon un message « trop petit » ou « trop grand » apparaîtra.

La ligne 530 incrémente le compteur de jeu (variable C) et remet à 0 le chiffre que vous avez utilisé (dans le tableau A()).

Si C = 9, on affiche la solution (variable N), puisqu'il ne reste plus de chiffre utilisable.

```
440 REM -----------------
450 REM SOLUTION
460 PRINT "NOMBRE PROPOSE : ";R
470 IF R=N THEN 570
480 IF R<N THEN 510
490 PRINT"C'EST TROP GRAND"
500 GOTO 520
510 PRINT"C'EST TROP PETIT"
520 LET A(S)=0
530 LET C=C+1
540 IF C=9 THEN 590
550 GOTO 160
560 REM -----------------
570 REM RESULTATS
580 PRINT "VOUS AVEZ TROUVEZ !"
590 PRINT "C'ETAIT ";N
```

NOMBRE PROPOSÉ : 45
C'EST TROP GRAND

Liste des variables

A()	Tableau des chiffres restants
C	Nombre d'essais
I	Indice de boucle FOR/NEXT
N	Le nombre secret
O$	Opération choisie
R	Résultat de la dernière opération
S	Chiffre choisi

SUPER NOMBRE SECRET - SUPER

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler

```
10 REM SUPER NOMBRE SECRET
20 REM --------------------
30 DIM A(9)
40 LET C=0
50 FOR I=1 TO 9
60 LET A(I)=1
70 NEXT I
80 LET N=INT(RND(1)*99)+1
90 PRINT "PREMIER CHIFFRE ?"
100 INPUT S
110 IF S<1 OR S>9 THEN 90
120 LET R=S
130 GOTO 460
140 REM --------------
150 REM LE JEU
160 PRINT "IL VOUS RESTE:"
170 FOR I=1 TO 9
180 IF A(I)=1 THEN PRINT I,
190 NEXT I
200 PRINT
210 PRINT "NOMBRE CHOISI:"
220 INPUT S
230 IF S<1 OR S>9 THEN 210
240 IF A(S)=0 THEN 160
250 PRINT "OPERATION (+,-,* OU /)"
260 INPUT O$
270 IF O$="+" THEN 320
280 IF O$="-" THEN 350
290 IF O$="*" THEN 380
300 IF O$="/" THEN 410
310 GOTO 160
320 IF R+S>99 THEN 160
330 LET R=R+S
340 GOTO 460
350 IF R-S<1 THEN 160
360 LET R=R-S
370 GOTO 460
380 IF R*S>99 THEN 160
390 LET R=R*S
400 GOTO 460
410 IF INT(R/S)<>R/S THEN 160
420 IF R/S<1 THEN 160
430 LET R=R/S
440 REM -----------------
450 REM SOLUTION
460 PRINT "NOMBRE PROPOSE : ";R
470 IF R=N THEN 570
480 IF R<N THEN 510
490 PRINT"C'EST TROP GRAND"
500 GOTO 520
510 PRINT"C'EST TROP PETIT"
520 LET A(S)=0
530 LET C=C+1
540 IF C=9 THEN 590
550 GOTO 160
560 REM -----------------
570 REM RESULTATS
580 PRINT "VOUS AVEZ TROUVEZ !"
590 PRINT "C'ETAIT ";N
```

LA PESÉE

Si vous continuez de vous servir de la balance de votre grand-mère, demandez-donc à votre ordinateur les poids qu'il convient de choisir pour procéder à une pesée. Mais attention aux tares cachées...

par Jacques Deconchat

Vous souvenez-vous des balances d'autrefois, les bonnes vieilles balances encore utilisées de nos jours, au moins dans les écoles ? Avec le système du double plateau, la grande précision importait peu, il suffisait que les masses de pesée soient justes. Cette fiche programme ne permettra pas de réaliser une vraie pesée, elle indiquera simplement les massses à utiliser.

Le programme est conçu pour la traditionnelle boîte de masses : 500, 200, 100, 50, 20, etc. Une petite restriction, toutefois : afin de simplifier, il n'y a que des masses entières. Mais il vous sera possible de le modifier à votre guise pour, par exemple, vérifier la justesse de vos conceptions en matière de pesée.

On pourrait concevoir une boîte plus astucieuse, permettant toutes les pesées avec encore moins de masses au départ ; il suffit de modifier les masses, enregistrées en DATA en fin de programme.

```
10  REM  PESEE
20  REM  -----
30  PRINT "L'ORDINATEUR DISPOSE D'UNE BOITE DE POIDS DE 1
    A 500 GRAMMES"
40  PRINT "VOUS POUVEZ LUI DEMANDER DE REALISER UNE PESEE"
50  PRINT "POUR N'IMPORTE QUELLE MASSE COMPRISE ENTRE
    1 GRAMME ET 1000 GRAMMES"
60  PRINT "ATTENTION : IL N'ACCEPTE QUE DES ENTIERS"
70  PRINT
80  PRINT "DONNEZ VOTRE MASSE"
90  INPUT M
```

Dans ce « préambule », l'ordinateur affiche un texte expliquant en quelques lignes ce que réalise le programme. La lecture se termine sur une question, en ligne 80 : « DONNEZ VOTRE MASSE ». Il convient de répondre par un nombre entier compris entre 0 et 1 000 – ce sont les limites extrêmes des possibilités de la boîte de masses.

```
100  IF M <  >  INT (M)
     OR M > 1000 THEN
     GOTO 80
110  PRINT
120  PRINT "IL VOUS
     FAUDRA LES MASSES
     SUIVANTES"
```

La ligne 100 permet de vérifier si la variable M, contenant la masse à peser, répond bien aux conditions imposées : variable entière, non supérieure à 1 000. Si ce n'est pas le cas, la question « DONNEZ VOTRE MASSE » sera reposée.

La ligne 120 est un titre servant à annoncer la réponse.

```
140  FOR I = 1 TO 13
150  READ B
160  IF M < B THEN  NEXT I
170  LET M = M - B
180  PRINT B;" ° ";
190  IF M <  > 0 THEN
     NEXT I
```

C'est ici que se trouve la principale partie du programme, c'est-à-dire l'algorithme de calcul.

Chaque masse de la boîte est comparée avec la masse introduite initialement. Si elle est trop grande, on la refuse pour prendre la suivante. Si elle est égale ou plus petite, on l'affiche et on la soustrait de la masse initiale. La ligne 190 permet de vérifier aisément si l'on a ou non terminé ; en effet, la différence, contenue dans M, sera nulle si l'on a fini.

```
210  PRINT "VOULEZ-VOUS
     UNE AUTRE
     PESEE ( O/N ) ?"
220  INPUT R$
230  IF R$ <  > "O" THEN
     END
240  RUN
```

C'est la fin du jeu, où l'on pose traditionnellement la question : « VOULEZ-VOUS REJOUER ? ». Une réponse "O" relance le programme.

La ligne 250 contient la liste des masses disponibles.

Sur le ZX 81, qui ne connaît pas les instructions READ et DATA, il sera plus simple d'enregistrer en début de programme les masses dans une table N(I) (avec N (1)=500, N(2)=200, N(3)=100, etc.). Il faudra alors remplacer la ligne 150 READ B par 150 LET B=N(1) pour que le programme fonctionne.

Liste des variables

M	masse à peser
I	variable de boucle
B	masse lue dans la boîte
R$	saisie de la réponse O ou N

LA PESÉE - LA PESÉE - LA PESÉE

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10  REM  PESEE
20  REM  -----
30  PRINT "L'ORDINATEUR DISPOSE D'UNE BOITE
    DE POIDS DE 1 A 500 GRAMMES"
40  PRINT "VOUS POUVEZ LUI DEMANDER DE
    REALISER UNE PESEE"
50  PRINT "POUR N'IMPORTE QUELLE MASSE
    COMPRISE ENTRE 1 GRAMME ET 1000 GRAMMES"
60  PRINT "ATTENTION : IL N'ACCEPTE QUE
    DES ENTIERS"
70  PRINT
80  PRINT "DONNEZ VOTRE MASSE"
90  INPUT M
100  IF M <  >  INT (M) OR M > 1000 THEN
     GOTO 80
110  PRINT
120  PRINT "IL VOUS FAUDRA LES MASSES
     SUIVANTES"
130  PRINT
140  FOR I = 1 TO 13
150  READ B
160  IF M < B THEN  NEXT I
170  LET M = M - B
180  PRINT B;" ° ";
190  IF M <  > 0 THEN  NEXT I
200  PRINT
210  PRINT "VOULEZ-VOUS UNE AUTRE PESEE
     ( O/N ) ?"
220  INPUT R$
230  IF R$ <  > "O" THEN  END
240  RUN
250  DATA    500,200,100,100,50,20,10,
     10,5,2,2,1,0
```



LABYRINTHIX

Il faut être aussi fou que Thésée pour imaginer d'aller se perdre volontairement dans un impossible labyrinthe, avec en prime le risque de tomber sur un horrible monstre affamé de chair et de sang. Notre labyrinthe ne vous posera pas autant de problèmes. La seule chose qui vous est demandée, c'est de dire où vous êtes, avec le moins d'essais possible. Ce ne sera pas toujours très facile.

par Jacques Deconchat

Vous êtes dans un labyrinthe qui comporte une sortie, visible sur votre plan. Vous pouvez vous déplacer en donnant une suite de lettres, avec la convention suivante : G pour aller à gauche, D pour aller à droite, H pour le haut et B pour le bas. Ainsi, à la question « votre déplacement », la réponse BBBB GGG HH, signifie que vous allez essayer de vous déplacer de quatre cases vers le bas, de trois vers la gauche et de deux vers le haut. Mais, malheureusement, vous ne savez pas où vous êtes au départ. Vous risquez donc de vous heurter à un mur. L'ordinateur vous indiquera alors de combien de cases vous avez pu vous déplacer sans toucher un mur.

La réponse « 6 cases libres » vous indique que, pour le déplacement précédent, seule la partie de trajet BBBB GG était possible.

Le programme, donné ici en Basic Microsoft standard, pourra poser quelques petits problèmes d'adaptation sur certains appareils : la ligne 12 est indispensable pour ceux qui ne disposeraient pas d'une fonction RANDOMIZE (en effet, le générateur aléatoire risque alors de donner toujours la même valeur, ce qui supprime l'intérêt du jeu).

La première partie du programme réalise le dessin d'un labyrinthe ; tout autre dessin sera possible, en changeant les lignes 30 à 150 ; sur les ZX, la ligne 20 (effacement d'écran et dimensionnement du tableau A$) devra être modifiée avec DIM A$ (13,21) (21 représente ici la longueur de la chaîne).

```
20 CLS:DIM A$(13)
30 A$(0)="  ABCDEFGHIJKLMNOPQR"
40 A$(1)=" ####################"
50 A$(2)="0#   #       #      #"
60 A$(3)="1# # # ####### #### #"
70 A$(4)="2# # #      ## #    #"
80 A$(5)="3# #    ### ## # ####"
90 A$(6)="4# ####   #    #  # #"
100 A$(7)="5  #    # # # ## #  #"
110 A$(8)="6# # # ## # #     # #"
120 A$(9)="7#   # #    # # ### #"
130 A$(10)="8# ### # # ## #     #"
140 A$(11)="9#       #    ##### #"
150 A$(12)=" ####################"
```

Confection du labyrinthe. Les 3 lignes 160, 170 et 180 réalisent le tirage aléatoire d'une position dans le labyrinthe, la ligne 180 permettant de vérifier que la case tirée est effectivement libre (sinon, on tire une autre case).

```
160 X=INT(RND*10+2)
170 Y=INT(RND*20+2)
180 IF MID$(A$(X),Y,1)="#" THEN 170
190 A=X:B=Y:T=0
```

La ligne 190 enregistre les deux valeurs tirées et met à 0 le compteur d'essai.

```
200 CLS:FOR I=0 TO 12:PRINT A$(I):NEXT I
210 PRINT:INPUT"VOTRE DEPLACEMENT :";R$
220 S=0
230 FOR I=1 TO LEN(R$)
240 X=X+(MID$(R$,I,1)="H")-(MID$(R$,I,1)
="B")
250 Y=Y+(MID$(R$,I,1)="G")-(MID$(R$,I,1)
="D")
260 IF MID$(A$(X),Y,1)="#" THEN 290
270 S=S+1
280 NEXT I                              ▶
```

```
▶
290 PRINT:PRINT"VOUS AVEZ ";S;" CASES LI
BRES"
300 INPUT"SOLUTION (O/N)";R$
310 IF LEFT$(R$,1)="N" THEN X=A:Y=B:T=T+
1:GOTO 200
```

C'est le « corps » du programme : l'affichage du labyrinthe est réalisé après un effacement d'écran par une simple boucle placée en ligne 200 ; la méthode utilisée permet de ne pas imposer de ligne ou de colonne particulière pour l'affichage, cela afin de faciliter l'adaptation sur plusieurs machines (seul l'effacement d'écran sera à modifier, en fonction de votre appareil).

Le décodage du déplacement est effectué par une boucle, de la ligne 230 à la ligne 280 ; la ligne 260 contrôle si la case considérée est libre ou non. Si ce n'est pas le cas, on part en ligne 290 pour afficher le nombre de cases libres.

Une réponse « non » à la question « solution (O/N) » provoque un nouveau dessin du labyrinthe et fait avancer le compteur d'essai d'une unité.

Si l'on répond « oui », on entre alors dans la partie fin du programme, de la ligne 320 à 350.

```
320 INPUT"LIGNE :";X:X=X+2
330 INPUT"COLONNE :";R$:Y=ASC(R$)-62
340 IF X=A AND Y=B THEN PRINT"BRAVO...";
T;" ESSAIS":END
350 PRINT"PERDU . C'ETAIT LIGNE ";A-2;",
 COLONNE ";CHR$(62+B)
```

Les lignes 320 et 330 demandent la ligne et la colonne : la ligne sera entrée sous forme d'un nombre, de 0 à 9. La colonne sous forme d'une lettre, de A à R ; elle est ensuite transformée en nombre.

La ligne 340 contrôle si l'on a gagné, et affiche dans ce cas le nombre d'essais effectués. Si l'on a perdu, le bon résultat est affiché par la ligne 350.

Le labyrinthe proposé a été limité à douze lignes et vingt colonnes pour permettre une transposition aisée sur plusieurs machines (certains affichages ne permettant pas d'aller au-delà de seize lignes) ; on pourra facilement modifier le programme pour élaborer des labyrinthes plus complexes, en tenant compte des possibilités de chaque appareil.

LABYRINTHIX-LABYRINTHIX-LAB

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10 REM LABYRINTHIX
12 INPUT"DONNEZ UN ENTIER ";J:FOR I=1 TO
 J:X=RND:NEXT I
20 CLS:DIM A$(13)
30 A$(0)="  ABCDEFGHIJKLMNOPQR"
40 A$(1)=" ####################"
50 A$(2)="0#   #       #      #"
60 A$(3)="1# # # ####### #### #"
70 A$(4)="2# # #      ## #    #"
80 A$(5)="3# #    ### ## # ####"
90 A$(6)="4# ####   #    #  # #"
100 A$(7)="5  #    # # # ##  # #"
110 A$(8)="6# # # ## # #     # #"
120 A$(9)="7#   # #    # # ### #"
130 A$(10)="8# ### # # ## #     #"
140 A$(11)="9#       #    ##### #"
150 A$(12)=" ####################"
160 X=INT(RND*10+2)
170 Y=INT(RND*20+2)
180 IF MID$(A$(X),Y,1)="#" THEN 170
190 A=X:B=Y:T=0
200 CLS:FOR I=0 TO 12:PRINT A$(I):NEXT I
210 PRINT:INPUT"VOTRE DEPLACEMENT :";R$
220 S=0
230 FOR I=1 TO LEN(R$)
240 X=X+(MID$(R$,I,1)="H")-(MID$(R$,I,1)
="B")
250 Y=Y+(MID$(R$,I,1)="G")-(MID$(R$,I,1)
="D")
260 IF MID$(A$(X),Y,1)="#" THEN 290
270 S=S+1
280 NEXT I
290 PRINT:PRINT"VOUS AVEZ ";S;" CASES LI
BRES"
300 INPUT"SOLUTION (O/N)";R$
310 IF LEFT$(R$,1)="N" THEN X=A:Y=B:T=T+
1:GOTO 200
320 INPUT"LIGNE :";X:X=X+2
330 INPUT"COLONNE :";R$:Y=ASC(R$)-62
340 IF X=A AND Y=B THEN PRINT"BRAVO...";
T;" ESSAIS":END
350 PRINT"PERDU . C'ETAIT LIGNE ";A-2;",
 COLONNE ";CHR$(62+B)
```



LES VERBES EN "ER"

Mais non, ce n'est pas fastidieux de réviser ses conjugaisons ! Cela peut même devenir très amusant si l'on suit ce petit programme. Il enseignera à votre enfant les finesses des terminaisons des verbes du premier groupe, à trois temps de l'indicatif.

par Jacques Deconchat

Le fonctionnement est très simple : on choisit un verbe du premier groupe (ceux qui se terminent en "er" à l'infinitif) que l'on entre dans l'appareil. On obtient alors l'affichage des quatre options possibles :

– 1 (présent),
– 2 (imparfait),
– 3 (futur),
– 4 (fin).

On tape alors, 1, 2, 3 ou 4, selon le cas.

```
10  REM  CONJUGAISON D'UN VERBE DU
    PREMIER GROUPE
20  PRINT "DONNEZ VOTRE VERBE"
30  INPUT V$
40  IF  LEN (V$) < 3 OR  RIGHT$
    (V$,2) <  > "ER" THEN  GOTO 20
50 R$ =  LEFT$ (V$, LEN (V$) - 2)
60  PRINT "1(PRESENT),2(IMPARFAIT)
    3(FUTUR) ,4(FIN)"
70  INPUT T
80  IF T <  > 1 AND T <  > 2
    AND T <  >  3 AND T <  >
    4 THEN  GOTO 60
90  ON T GOTO 100,200,300,400
```

Cette partie s'occupe de l'introduction du verbe qui sera enregistré dans une mémoire V$. On vérifie qu'il a plus de trois lettres et qu'il se termine bien par "er" (ligne 40).
Si ce n'est pas le cas, il faut alors demander un autre verbe. Puis le radical (obtenu en retranchant la terminaison "er") est enregistré dans une mémoire R$.
Le programme demande ensuite (ligne 60) le temps désiré. On devra alors répondre 4 si l'on désire pas continuer à conjuguer. La réponse est enregistrée dans une variable T, la ligne 80 servant à vérifier que la valeur frappée correspond bien aux seules options proposées.
La ligne 90 (aiguillage) dirige alors sur la ligne correspondant à l'option choisie, en fonction de la valeur de T.

```
100  REM  CONJUGAISON
     AU PRESENT
110  PRINT "PRESENT DE ";V$
120  PRINT "-------------"
130  PRINT "JE ";R$;"E"
140  PRINT "TU ";R$;"ES"
150  PRINT "IL ";R$;"E"
160  PRINT "NOUS ";R$;"ONS"
170  PRINT "VOUS ";R$;"EZ"
180  PRINT "ILS ";R$;"ENT"
190  GOTO 10
```

Le verbe est conjugué au présent, en utilisant le radical conservé dans R$. Un titre « PRESENT DU VERBE » est affiché. Lorsque la conjugaison est effectuée, on retourne poser la question initiale : « DONNEZ VOTRE VERBE ».

```
200  REM  CONJUGAISON A L'IMPARFAIT
210  PRINT "IMPARFAIT DE ";V$
220  PRINT "------------------------
230  PRINT "JE ";R$;"AIS"
240  PRINT "TU ";R$;"AIS"
250  PRINT "IL ";R$;"AIT"
260  PRINT "NOUS ";R$;"IONS"
270  PRINT "VOUS ";R$;"IEZ"
280  PRINT "ILS ";R$;"AIENT"
290  GOTO 10
300  REM  CONJUGAISON AU FUTUR
310  PRINT "FUTUR DE ";V$
320  PRINT "------------------------
330  PRINT "JE ";R$;"ERAI"
340  PRINT "TU ";R$;"ERAS"
350  PRINT "IL ";R$;"ERA"
360  PRINT "NOUS ";R$;"ERONS"
370  PRINT "VOUS ";R$;"EREZ"
380  PRINT "ILS ";R$;"ERONT"
390  GOTO 10
```

Ces deux parties vont réaliser la conjugaison à l'imparfait et au futur, en suivant la même technique que précédemment. Seules les terminaisons changent (on pourrait d'ailleurs simplifier le programme en tenant compte de cette remarque).
Il faut remarquer que le programme proposé recèle, pour les questions de longueur, une petite imperfection : en effet, si le verbe à conjuguer commence par une voyelle, la première personne sera incorrecte (on ne dit pas « je aime », mais « j'aime »). A vous de remédier à cette lacune en introduisant un test sur la première lettre du verbe introduit.
La ligne 400, bien sûr, est la ligne de fin de programme. Pour le ZX 81 ou le ZS Spectrum, il conviendra de modifier les lignes 40, 50 et 90 de la façon suivante :
en ligne 40, remplacer RIGHT$(V$,2) par V$(LEN(V$)-2 TO)
en ligne 50, remplacer LEFT$(V$,LEN(V$)-2) par V$(TO LEN(V$)-2)
la ligne 90 enfin sera simplement remplacée par GOTO T ✶ 100.

Liste des variables

V$ contient le verbe à conjuguer
R$ contient le radical du verbe
V variable permettant l'aiguillage

LES VERBES EN "ER" - LES VER

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10  REM  CONJUGAISON D'UN VERBE DU PREMIER GROUPE
20  PRINT "DONNEZ VOTRE VERBE"
30  INPUT V$
40  IF  LEN (V$) < 3 OR  RIGHT$ (V$,2) <  > "ER"
    THEN  GOTO 20
50 R$ =  LEFT$ (V$, LEN (V$) - 2)
60  PRINT "1(PRESENT) ,2(IMPARFAIT) ,3(FUTUR) ,
    4(FIN)"
70  INPUT T
80  IF T <  > 1 AND T <  > 2 AND T <  > 3 AND T
    <  > 4 THEN  GOTO 60
90  ON T GOTO 100,200,300,400
100  REM  CONJUGAISON AU PRESENT
110  PRINT "PRESENT DE ";V$
120  PRINT "-------------------------"
130  PRINT "JE ";R$;"E"
140  PRINT "TU ";R$;"ES"
150  PRINT "IL ";R$;"E"
160  PRINT "NOUS ";R$;"ONS"
170  PRINT "VOUS ";R$;"EZ"
180  PRINT "ILS ";R$;"ENT"
190  GOTO 10
200  REM  CONJUGAISON A L'IMPARFAIT
210  PRINT "IMPARFAIT DE ";V$
220  PRINT "-------------------------"
230  PRINT "JE ";R$;"AIS"
240  PRINT "TU ";R$;"AIS"
250  PRINT "IL ";R$;"AIT"
260  PRINT "NOUS ";R$;"IONS"
270  PRINT "VOUS ";R$;"IEZ"
280  PRINT "ILS ";R$;"AIENT"
290  GOTO 10
300  REM  CONJUGAISON AU FUTUR
310  PRINT "FUTUR DE ";V$
320  PRINT "-------------------------"
330  PRINT "JE ";R$;"ERAI"
340  PRINT "TU ";R$;"ERAS"
350  PRINT "IL ";R$;"ERA"
360  PRINT "NOUS ";R$;"ERONS"
370  PRINT "VOUS ";R$;"EREZ"
380  PRINT "ILS ";R$;"ERONT"
390  GOTO 10
400  END
```



SUITES LOGIQUES

Une suite de trois chiffres sur votre écran. Vous désirez connaître le quatrième ? Un raisonnement logique vous le donnera.

par Jean-François Sehan

Sans prétendre à la comparaison avec un test poussé de quotient intellectuel, ce programme de suites logiques peut constituer un bon entraînement au raisonnement.
Son but est de trouver le nombre découlant d'une suite affichée à l'écran. Pour cela, il propose trois nombres dans un ordre donné, obéissant à un algorithme utilisant l'addition, la soustraction ou la multiplication. A vous de trouver les nombres suivants.

```
************************
SUITE : 1 3 19
LE QUATRIÈME CHIFFRE ?
? 66
SUITE : 1 3 19 147
LE CINQUIÈME CHIFFRE ?
? 1542
PERDU !
SUITE : 1 3 19 147 1171
SIXIÈME CHIFFRE : 9363
(1171 * 8)+(−5)
```

La suite logique étant placée dans le tablea A(1) à A(6), A(1) prend la valeur 1 dès le début du jeu.

```
10 REM SUITES LOGIQUES
20 REM ---------------
30 DIM A(6)
40 LET A(1)=1
```

Les lignes 50 à 80 placent dans les variables I, J, K et L quatre nombres aléatoires qui détermineront la suite logique.

```
50 LET I=INT(RND(1)*10)+1
60 LET J=INT(RND(1)*10)+1
70 LET K=INT(RND(1)*10)+1
80 LET L=INT(RND(1)*10)+1
```

Les lignes 90 à 170 calculent la suite logique (6 chiffres au total).
La ligne 230 saisit votre première solution (variable S). Si S=0, on retourne à l'affichage. Si votre proposition est bonne, la ligne 250, après un test, envoie en ligne 410 pour un message de félicitations.

```
90 IF K>5 THEN LET J=-J
100 IF L>5 THEN 150
110 FOR U=2 TO 6
120 LET A(U)=A(U-1)*I+J
130 NEXT U
140 GOTO 200
150 FOR U=2 TO 6
160 LET A(U)=(A(U-1)+J)*I
170 NEXT U
180 REM -----------------------
190 REM VOTRE SOLUTION
200 PRINT "************************"
210 PRINT "SUITE:";A(1);A(2);A(3)
220 PRINT"LE QUATRIEME CHIFFRE ?"
230 INPUT S
240 IF S=0 THEN 210
250 IF S=A(4) THEN 410
```

```
************************
SUITE : 1 − 12 − 38
LE QUATRIÈME CHIFFRE ?
? −96
```

Dans le cas contraire, la même suite est affichée avec un chiffre supplémentaire (ligne 260). La ligne 280 saisit votre deuxième solution (même méthode qu'en ligne 230).
Après avoir testé votre proposition, la ligne 300 renvoie en 410 si celle-ci est juste.

```
260 PRINT "SUITE:";A(1);A(2);A(3);A(4)
270 PRINT "LE CINQUIEME CHIFFRE ?"
280 INPUT S
290 IF S=0 THEN 260
300 IF S=A(5) THEN 410
```

```
SUITE : 1 − 12 − 38 − 90
LE CINQUIÈME CHIFFRE ?
? −186
```

Sinon les lignes 320 à 380 affichent la bonne solution.

```
310 REM -----------------------
320 PRINT"PERDU !"
330 PRINT "SUITE:";A(1);A(2);A(3);A(4);A(5)
340 PRINT "SIXIEME CHIFFRE:";A(6)
350 IF L>5 THEN 380
360 PRINT "(";A(5);"*";I;")+(";J;")"
370 GOTO 390
380 PRINT "(";A(5);"+(";J;"))*";I
390 GOTO 50
400 REM -----------------------
410 PRINT"GAGNE !"
420 GOTO 50
```

```
PERDU !
SUITE : 1 − 12 − 38 − 90 − 194
SIXIÈME CHIFFRE : −402
(−194+(−7)) * 2
```

Liste des variables

A()	Contient la suite logique
I, J, K et L	Nombres aléatoires pour le calcul de la suite logique
S	Solution proposée
U	Indice de boucle FOR/NEXT

SUITES LOGIQUES - SUITES LO

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10 REM SUITES LOGIQUES
20 REM -------------------------
30 DIM A(6)
40 LET A(1)=1
50 LET I=INT(RND(1)*10)+1
60 LET J=INT(RND(1)*10)+1
70 LET K=INT(RND(1)*10)+1
80 LET L=INT(RND(1)*10)+1
90 IF K>5 THEN LET J=-J
100 IF L>5 THEN 150
110 FOR U=2 TO 6
120 LET A(U)=A(U-1)*I+J
130 NEXT U
140 GOTO 200
150 FOR U=2 TO 6
160 LET A(U)=(A(U-1)+J)*I
170 NEXT U
180 REM -------------------------
190 REM VOTRE SOLUTION
200 PRINT "*************************"
210 PRINT "SUITE:";A(1);A(2);A(3)
220 PRINT"LE QUATRIEME CHIFFRE ?"
230 INPUT S
240 IF S=0 THEN 210
250 IF S=A(4) THEN 410
260 PRINT "SUITE:";A(1);A(2);A(3);A(4)
270 PRINT "LE CINQUIEME CHIFFRE ?"
280 INPUT S
290 IF S=0 THEN 260
300 IF S=A(5) THEN 410
310 REM -------------------------
320 PRINT"PERDU !"
330 PRINT "SUITE:";A(1);A(2);A(3);A(4);A(5)
340 PRINT "SIXIEME CHIFFRE:";A(6)
350 IF L>5 THEN 380
360 PRINT "(";A(5);"*";I;")+(";J;")"
370 GOTO 390
380 PRINT "(";A(5);"+(";J;"))*";I
390 GOTO 50
400 REM -------------------------
410 PRINT"GAGNE !"
420 GOTO 50
```

L'ASTROLOGIE

Bélier ? Lion ? Scorpion ? Ou peut-être Sagittaire ? Entrez une date de naissance, et ce programme nous dira tout, du signe zodiacal correspondant aux traits généraux caractérisant les natifs de ce signe.

par Jean-François Sehan

L'appel du sous-programme en ligne 330 place dans le tableau J() le nombre de jours de chaque mois, dans P() le nombre de jours entre deux signes, dans S$() les douze signes du zodiaque et dans D$() la dominante de chacun de ces signes.

```
310 REM -------------------------------
320 REM DONNEES
330 FOR I=1 TO 12
340 READ J(I),P(I),S$(I),D$(I)
350 NEXT I
360 LET P(13)=29
370 RETURN
380 DATA 31,20,VERSEAU,"D'AIR. ALTRUISTE, SENSIBLE, INDEPENDANT, ANTICONFOR
 MISTE ET LUCIDE, IL CHERCHE A SE REALISER DANS LE DOMAINE MORAL OU SOCIAL."
390 DATA 28,30,POISSONS,"D'EAU. SECRET, IMAGINATIF, EMOTIF, IMPRESSIONNABLE
 ET SENSIBLE, IL SE VEUT PARTIE INTEGRANTE DE L'UNIVERS."
400 DATA 31,30,BELIER,"DE FEU. HOMME OU FEMME D'ACTION,PLEIN DE VITALITE ET
 DE  TONUS, IL AIME L'EFFORT."
410 DATA 30,30,TAUREAU,"DE TERRE. PASSIONNE, CHARNEL, SENSUEL. IL FAIT
 PREUVE D'UNE ACTIVITE TENACE."
420 DATA 31,31,GEMEAUX,"D'AIR. TRES VIF, DOUE, IL MANIFESTE UNE GRANDE
 FACULTE  D'ADAPTATION ET BEAUCOUP DE RAPIDITE DANS SON TRAVAIL."
430 DATA 30,32,CANCER,"D'EAU. DOUX, CALME, CHARMEUR, IL AIME FLANER, REVER,
 SE  REFUGIER DANS L'IMAGINAIRE."
440 DATA 31,31,LION,"DE FEU.VOLONTAIRE,GENEREUX,ACTIF,IL POSSEDE LE GOUT DE
 SE REALISER DANS LES GRANDES ENTREPRISES.
450 DATA 31,31,VIERGE,"DE TERRE. METHODIQUE,METICULEUX, CURIEUX, IL RECHERCHE
LE CALME DE L'ESPRIT ET LA SECURITE."
460 DATA 30,31,BALANCE,"D'AIR. OUVERT,SENSIBLE, AFFECTUEUX, SUBTIL, INTUITIF,
 IL EST A LA RECHERCHE DE SON EQUILIBRE."
470 DATA 31,30,SCORPION,"D'EAU. PASSIONNE. COMBATIF, ALTRUISTE, C'EST UN TRA
 VAILLEUR ACHARNE."
480 DATA 30,30,SAGITTAIRE,"DE FEU. BRILLANT,EQUILIBRE, SERIEUX, IL EPROUVE LE
 BESOIN DE FRANCHIR SES LIMITES MATERIELLES ET SPIRITUELLES."
490 DATA 31,30,CAPRICORNE,"DE TERRE. FLEGMATIQUE, PASSIONNE, REFLECHI ET
 METHODIQUE, IL EST AUSSI PROFOND, GRAVE ET DISCIPLINE."
```

Si votre ordinateur ne possède pas les instructions DATA et READ, il faut remplacer les lignes 330 à 490 par :

330 J(1)=31
340 P(1)=20
350 S$(1)=« VERSEAU »
360 D$(1)=« D'AIR. ALTRUISTE,... ETC
370 J(2)=28
etc., en prenant les données des DATA des lignes 380 à 490.

Puis les lignes 80 à 130 saisissent le jour et le mois de naissance et calculent le nombre de jours vécus depuis le 1er janvier (variable N).

```
10 REM ASTROLOGIE
20 REM ------------------------------
30 DIM P(13)
40 DIM J(12)
50 DIM S$(12)
60 DIM D$(12)
70 GOSUB 330
80 PRINT"JOUR ET MOIS DE NAISSANCE"
90 INPUT J,M
100 FOR I=1 TO M-1
110 LET N=N+J(I)
120 NEXT I
130 LET N=N+J
```

Ce sont les lignes 160 à 230 qui déterminent le signe zodiacal, en comparant le nombre de jours vécus N avec la somme des écarts entre chaque signe (renseignements pris dans le tableau P).

```
140 REM ------------------
150 REM CALCUL DU SIGNE
160 LET S=1
170 LET C=C+P(S)
180 IF N<C THEN 210
190 LET S=S+1
200 GOTO 170
210 LET S=S-1
220 IF S>0 THEN 240
230 LET S=12
```

Il ne reste plus, aux lignes 260 à 290, qu'à afficher le signe du tableau S$() ainsi que la dominante contenue dans le tableau D$().

```
240 REM --------------------
250 REM SIGNE ASTROLOGIQUE
260 PRINT:PRINT "-ASTROLOGIE
    -": PRINT
270 PRINT"VOTRE SIGNE:";S$(S)
280 PRINT:PRINT"DOMINANTE:"
290 PRINT "SIGNE ";D$(S)
300 END
```

```
JOUR ET MOIS DE NAISSANCE
? 4,1
-ASTROLOGIE-
VOTRE SIGNE : CAPRICORNE
DOMINANTE :
SIGNE DE TERRE. FLEGMA
TIQUE, PASSIONNE, REFLECHI
ET METHODIQUE, IL EST AUSSI
PROFOND, GRAVE ET
DISCIPLINE.
```

Liste des variables

C	total du nombre de jours d'écart entre les signes
D$()	liste des dominantes de chaque signe
I	indice de boucle FOR/NEXT
J	jour de naissance
J()	nombre de jours par mois
M	mois de naissance
N	nombre de jours vécus depuis le 1er janvier
P()	tableau des écarts entre deux signes
S	numéro du signe
S$()	liste des douze signes du zodiaque

L'ASTROLOGIE - L'ASTROLOGIE - L'A

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10 REM ASTROLOGIE
20 REM -------------------------------
30 DIM P(13)
40 DIM J(12)
50 DIM S$(12)
60 DIM D$(12)
70 GOSUB 330
80 PRINT"JOUR ET MOIS DE NAISSANCE"
90 INPUT J,M
100 FOR I=1 TO M-1
110 LET N=N+J(I)
120 NEXT I
130 LET N=N+J
140 REM ------------------------------
150 REM CALCUL DU SIGNE
160 LET S=1
170 LET C=C+P(S)
180 IF N<C THEN 210
190 LET S=S+1
200 GOTO 170
210 LET S=S-1
220 IF S>0 THEN 240
230 LET S=12
240 REM ------------------------------
250 REM SIGNE ASTROLOGIQUE
260 PRINT:PRINT "-ASTROLOGIE-":PRINT
270 PRINT"VOTRE SIGNE:";S$(S)
280 PRINT:PRINT"DOMINANTE:"
290 PRINT "SIGNE ";D$(S)
300 END
310 REM ------------------------------
320 REM DONNEES
330 FOR I=1 TO 12
340 READ J(I),P(I),S$(I),D$(I)
350 NEXT I
360 LET P(13)=29
370 RETURN
380 DATA 31,20,VERSEAU,"D'AIR. ALTRUISTE, SENSIBLE, INDEPENDANT, ANTI
 CONFORMISTE ET LUCIDE, IL CHERCHE A SE REALISER DANS LE DOMAINE
 MORAL OU SOCIAL."
390 DATA 28,30,POISSONS,"D'EAU. SECRET, IMAGINATIF, EMOTIF, IMPRES
 SIONNABLE ET SENSIBLE, IL SE VEUT PARTIE INTEGRANTE DE L'UNIVERS."
400 DATA 31,30,BELIER,"DE FEU. HOMME OU FEMME D'ACTION, PLEIN DE VITA
 LITE ET DE TONUS, IL AIME L'EFFORT."
410 DATA 30,30,TAUREAU,"DE TERRE. PASSIONNE, CHARNEL, SENSUEL. IL
 FAIT PREUVE D'UNE ACTIVITE TENACE."
420 DATA 31,31,GEMEAUX,"D'AIR. TRES VIF, DOUE,IL MANIFESTE UNE GRANDE
 FACULTE D'ADAPTATION ET BEAUCOUP DE RAPIDITE DANS SON TRAVAIL."
430 DATA 30,32,CANCER,"D'EAU. DOUX, CALME, CHARMEUR, IL AIME FLANER,
 REVER, SE REFUGIER DANS L'IMAGINAIRE."
440 DATA 31,31,LION,"DE FEU. VOLONTAIRE, GENEREUX, ACTIF, IL POSSEDE
 LE GOUT DE . SE REALISER DANS LES GRANDES ENTREPRISES.
450 DATA 31,31,VIERGE,"DE TERRE. METHODIQUE, METICULEUX, CURIEUX, IL
 RECHERCHE LE CALME DE L'ESPRIT ET LA SECURITE."
460 DATA 30,31,BALANCE,"D'AIR. OUVERT,SENSIBLE, AFFECTUEUX, SUBTIL,
 INTUITIF, IL EST A LA RECHERCHE DE SON EQUILIBRE."
470 DATA 31,30,SCORPION,"D'EAU.PASSIONNE, COMBATIF, ALTRUISTE, C'EST
 UN TRAVAILLEUR ACHARNE."
480 DATA 30,30,SAGITTAIRE,"DE FEU. BRILLANT,EQUILIBRE, SERIEUX, IL E
 PROUVE LE BESOIN DE FRANCHIR SES LIMITES MATERIELLES ET SPIRITUELLES."
490 DATA 31,30,CAPRICORNE,"DE TERRE. FLEGMATIQUE, PASSIONNE, REFLECHI
 ET METHODIQUE, IL EST AUSSI PROFOND, GRAVE ET DISCIPLINE."
```

LES CRYPTOGRAMMES

Si vous aimez les messages secrets, les informations codées, les encres sympathiques, bref, si vous êtes féru de cryptographie, vous pourrez donner libre cours à votre passion en échangeant des textes indécodables... sauf pour celui qui en connaît la clef !

par Jean-François Sehan

En effet, le programme vous demande s'il s'agit d'un message à coder ou à décoder, puis le mot de passe (la clef du message). Vous devez taper un message quelconque, par exemple « VOTRE ORDINATEUR » ou « DE NORBERT A MARCEL». Sans cette clef, il n'est pas possible de décoder un message.

```
CODAGE OU DECODAGE (C/D)
? C
MOT DE PASSE
? IF THEN ELSE
MESSAGE
? VOTRE ORDINATEUR
MESSAGE CODE :
MRCOEROWAFZCENO
```

La ligne 50 saisit le type d'opération à effectuer : codage ou décodage. Les lignes 60 et 70 envoient aux programmes concernés.

```
10 REM CRYPTOGRAMMES
20 REM -------------------------
30 DIM C(26)
40 PRINT"CODAGE
   OU DECODAGE (C/D)"
50 INPUT R$
60 IF R$="D" THEN 190
70 IF R$<>"C" THEN 40
```

Codages : Après l'appel du sous-programme de saisie du mot de passe, du message et de la création du tableau de transcription, on code une à une toutes les lettres du message M$ à l'aide d'une boucle FOR/NEXT. Seules les lettres sont réellement codées, les autres signes sont ignorés. Il faut remplacer l'instruction MID$(M$,I,1) par M$(I) pour le ZX 81 et le Spectrum, et par SEG$(M$,I,1) pour le TI 99/4A (lignes 130 et 140).

```
80 REM -------------------------
90 REM CODAGE
100 GOSUB 310
110 PRINT"MESSAGE CODE:"
120 FOR I=1 TO LEN(M$)
130 IF MID$(M$,I,1)<"A" OR MID$
    (M$,I,1)>"Z" THEN 150
140 PRINT CHR$(C(ASC(MID$
    (M$,I,1))-64)+64);
150 NEXT I
160 END
```

Décodage : Comme pour le codage, on fait d'abord appel au sous-programme en 310, puis on décode une à une toutes les lettres du message M$ à l'aide d'une boucle FOR/NEXT. Il faut aussi remplacer les MID$ des lignes 220 et 240. →

```
170 REM -------------------------
180 REM DECODAGE
190 GOSUB 310
200 PRINT"MESSAGE DECODE:"
210 FOR I=1 TO LEN(M$)
220 IF MID$(M$,I,1)<"A" OR
    MID$(M$,I,1)>"Z" THEN 270
230 FOR J=1 TO 26
240 IF ASC(MID$(M$,I,1))
    -64<>C(J) THEN 260
250 PRINT CHR$(J+64);
260 NEXT J
270 NEXT I
280 END
```

Les lignes 310 à 490 créent dans le tableau C() les codes de transcription. Dès que le mot de passe est saisi, on place dans le tableau C() la correspondance de chaque lettre en fonction du mot de passe P$. Un exemple : si C(1)=3, la lettre A devient C ; si C(26)=5, la lettre Z devient E ; A étant la 1re lettre de l'alphabet, C la 3e, Z la 26e et E la 5e. Les instructions MID$(P$,I,1) seront remplacées par P$(I) sur les Sinclair et par SEG$(P$,I,1) sur les Texas Instruments.

```
290 REM ------------------------------
300 REM CREATION CODES
310 PRINT"MOT DE PASSE"
320 INPUT P$
330 IF P$="" THEN 310
340 LET D=1
350 FOR I=1 TO LEN(P$)
360 IF MID$(P$,I,1)<"A" OR MID$(P$,I,1)>"Z" THEN 400
370 IF C(ASC(MID$(P$,I,1))-64)<>0 THEN 400
380 LET C(ASC(MID$(P$,I,1))-64)=D
390 LET D=D+1
400 NEXT I
410 FOR I=26 TO 1 STEP -1
420 IF C(I)<>0 THEN 450
430 LET C(I)=D
440 LET D=D+1
450 NEXT I
460 PRINT"MESSAGE"
470 INPUT M$
480 IF M$="" THEN 460
490 RETURN
```

Liste des variables

C()	tableau de transcription
D	nombre de lettres déjà dans le tableau
I	indice de boucle FOR/NEXT
J	indice de boucle FOR/NEXT
M$	message à coder ou à décoder
P$	mot de passe
R$	choix de l'utilisateur

LES CRYPTOGRAMMES - LES CRYPTO

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10 REM CRYPTOGRAMMES
20 REM --------------------------------
30 DIM C(26)
40 PRINT"CODAGE OU DECODAGE (C/D)"
50 INPUT R$
60 IF R$="D" THEN 190
70 IF R$<>"C" THEN 40
80 REM --------------------------------
90 REM CODAGE
100 GOSUB 310
110 PRINT"MESSAGE CODE:"
120 FOR I=1 TO LEN(M$)
130 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" THEN 150
140 PRINT CHR$(C(ASC(MID$(M$,I,1))-64)+64);
150 NEXT I
160 END
170 REM --------------------------------
180 REM DECODAGE
190 GOSUB 310
200 PRINT"MESSAGE DECODE:"
210 FOR I=1 TO LEN(M$)
220 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" THEN 270
230 FOR J=1 TO 26
240 IF ASC(MID$(M$,I,1))-64<>C(J) THEN 260
250 PRINT CHR$(J+64);
260 NEXT J
270 NEXT I
280 END
290 REM --------------------------------
300 REM CREATION CODES
310 PRINT"MOT DE PASSE"
320 INPUT P$
330 IF P$="" THEN 310
340 LET D=1
350 FOR I=1 TO LEN(P$)
360 IF MID$(P$,I,1)<"A" OR MID$(P$,I,1)>"Z" THEN 400
370 IF C(ASC(MID$(P$,I,1))-64)<>0 THEN 400
380 LET C(ASC(MID$(P$,I,1))-64)=D
390 LET D=D+1
400 NEXT I
410 FOR I=26 TO 1 STEP -1
420 IF C(I)<>0 THEN 450
430 LET C(I)=D
440 LET D=D+1
450 NEXT I
460 PRINT"MESSAGE"
470 INPUT M$
480 IF M$="" THEN 460
490 RETURN
```

LES TOURS DE HANOI

Faire passer sur la tour de droite, et dans le même ordre, des disques empilés sur la tour de gauche, en vous aidant de la tour du milieu... un jeu qui demande patience et réflexion.

par Jean-François Sehan

Vous pouvez choisir le nombre de disques à manœuvrer, entre trois et six.
Le programme demande la tour de départ du mouvement, puis la tour d'arrivée : entrer 1, 2 ou 3 numéros (des tours). Il refuse les ordres impossibles à exécuter : on ne peut poser un disque sur un autre moins large. Mais un disque peut aller de la tour 1 à la tour 3, puis revenir à la tour 2. Il affiche aussi tout au long du jeu le nombre d'essais.
Un petit conseil : il faut mieux pour vos nerfs commencer par trois disques !
La ligne 40 saisit le nombre de disques à jouer (trois à sept). Puis on dimensionne un tableau D pour connaître pendant le jeu l'emplacement des disques sur les trois tours. On remplit le tableau de départ en mettant tous les disques sur la première tour, le disque le plus grand étant placé à la position 1.

```
10 REM TOURS DE HANOI
20 REM ----------------------
30 PRINT"NOMBRE DE DISQUES"
40 INPUT N
50 IF N<3 OR N>7 THEN 30
60 DIM D(3,7)
70 FOR I=1 TO 7
80 IF I>N THEN 100
90 LET D(1,I)=8-I
100 NEXT I
110 LET E=0
```

Les lignes 150 à 210 saisissent le numéro de la tour de départ (on déplace dans ce cas le disque supérieur).

```
120 REM -------------------
130 REM LE JEU
140 GOSUB 510
150 PRINT"TOUR DE DEPART:"
160 INPUT TD
170 IF TD<1 OR TD>3 THEN 150
180 LET J=TD
190 GOSUB 430
200 IF Q=1 THEN 150
210 LET DD=I
```

Le sous-programme en 430 trouve le numéro de ce disque. Si cette tour ne comporte aucun disque, on met le drapeau à 1 (variable Q). Au retour du sous-programme, si Q=1 on revient à la saisie de la tour de départ.

```
410 REM -----------------
420 REM NUMERO DU DISQUE
    SUPERIEUR
430 LET Q=0
440 FOR I=7 TO 1 STEP -1
450 IF D(J,I)<>0 THEN 480
460 NEXT I
470 LET Q=1
480 RETURN
```

On utilise le même principe pour la tour d'arrivée. Au retour du sous-programme en 430, on teste si le disque du dessus n'est pas plus petit , auquel cas on reviendrait à la saisie de la tour de départ.

```
220 PRINT"TOUR D'ARRIVEE:"
230 INPUT TA
240 IF TA<1 OR TA>3 THEN 220
250 IF TA=TD THEN 150
260 LET J=TA
270 GOSUB 430
280 IF Q=1 THEN 300
290 IF D(TD,DD)>D(TA,I)
    THEN 150
300 LET DA=I+1
310 LET D(TA,DA)=D(TD,DD)
320 LET D(TD,DD)=0
330 LET E=E+1
340 PRINT"ESSAIS:";E
```

Les lignes 350 à 370 testent sur la troisième tour si tous les disques sont déjà placés. Si ce n'est pas le cas, on retourne à la ligne 140 pour l'essai suivant, sinon on affiche le message « REUSSI » et le nombre d'essais.

```
350 FOR I=1 TO N
360 IF D(3,I)=0 THEN 140
370 NEXT I
380 GOSUB 510
390 PRINT"REUSSI EN";E;"COUPS"
400 END
```

C'est le sous-programme des lignes 510 à 650 qui dessine sur l'écran les trois tours à partir du tableauD().

```
490 REM ------------------------
500 REM AFFICHE LES DISQUES
510 PRINT"TOUR 1 TOUR 2 TOUR 3"
520 FOR J=7 TO 1 STEP -1
530 FOR I=1 TO 3
540 FOR K=7 TO 1 STEP -1
550 IF K>D(I,J) THEN 580
560 PRINT "*";
570 GOTO 590
580 PRINT" ";
590 NEXT K
600 PRINT"I     ";
610 NEXT I
620 PRINT
630 NEXT J
640 PRINT
650 RETURN
```

```
      TOUR 1     TOUR 2     TOUR 3
            I          I          I
            I          I          I
            I          I          I
            I          I          I
       *****I          I         *I
      ******I          I        **I
     *******I      ****I       ***I
TOUR DE DEPART :
? 3
TOUR D'ARRIVEE :
? 2
ESSAIS : 11
```

Liste des variables

D()	emplacement des disques sur les tours
DA	emplacement du disque sur la tour d'arrivée
DD	emplacement du disque sur la tour de départ
E	nombre d'essais
I	indice de boucle FOR/NEXT
J	indice de boucle FOR/NEXT
K	indice de boucle FOR/NEXT
N	nombre de disques
Q	si Q=1, il y a erreur de déplacement
TA	tour d'arrivée
TD	tour de départ

LES TOURS DE HANOI - LES TOURS DE

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler

```
10 REM TOURS DE HANOI
20 REM ------------------------------
30 PRINT"NOMBRE DE DISQUES"
40 INPUT N
50 IF N<3 OR N>7 THEN 30
60 DIM D(3,7)
70 FOR I=1 TO 7
80 IF I>N THEN 100
90 LET D(1,I)=8-I
100 NEXT I
110 LET E=0
120 REM ----------------------------
130 REM LE JEU
140 GOSUB 510
150 PRINT"TOUR DE DEPART:"
160 INPUT TD
170 IF TD<1 OR TD>3 THEN 150
180 LET J=TD
190 GOSUB 430
200 IF Q=1 THEN 150
210 LET DD=I
220 PRINT"TOUR D'ARRIVEE:"
230 INPUT TA
240 IF TA<1 OR TA>3 THEN 220
250 IF TA=TD THEN 150
260 LET J=TA
270 GOSUB 430
280 IF Q=1 THEN 300
290 IF D(TD,DD)>D(TA,I) THEN 150
300 LET DA=I+1
310 LET D(TA,DA)=D(TD,DD)
320 LET D(TD,DD)=0
330 LET E=E+1
340 PRINT"ESSAIS:";E
350 FOR I=1 TO N
360 IF D(3,I)=0 THEN 140
370 NEXT I
380 GOSUB 510
390 PRINT"REUSSI EN";E;"COUPS"
400 END
410 REM ----------------------------
420 REM NUMERO DU DISQUE SUPERIEUR
430 LET Q=0
440 FOR I=7 TO 1 STEP -1
450 IF D(J,I)<>0 THEN 480
460 NEXT I
470 LET Q=1
480 RETURN
490 REM ----------------------------
500 REM AFFICHE LES DISQUES
510 PRINT"TOUR 1       TOUR 2       TOUR 3"
520 FOR J=7 TO 1 STEP -1
530 FOR I=1 TO 3
540 FOR K=7 TO 1 STEP -1
550 IF K>D(I,J) THEN 580
560 PRINT "*";
570 GOTO 590
580 PRINT" ";
590 NEXT K
600 PRINT"I     ";
610 NEXT I
620 PRINT
630 NEXT J
640 PRINT
650 RETURN
```



LE MORSE

Futurs radios amateurs, voilà le plus court chemin pour devenir un bon « pianiste »

par Jean-François Sehan

Au lancement du programme, un menu vous est proposé : 1 ALPHABET 1.2 MESSAGE 2. Vous choisissez une option en en tapant le numéro.

Alphabet : L'alphabet complet s'affiche sur votre écran en code morse ; puis le programme retourne au menu.

Message : Vous entrez votre message puis, en tapant sur ENTER (RETURN ou ENTREE sur d'autres machines), vous obtenez sa traduction en code morse. Vous pouvez ensuite soit redemander un message, soit retaper sur ENTER pour revenir au menu.

```
1 ALPHABET
2 MESSAGE
VOTRE CHOIX
? 2
MESSAGE
? VOTRE ORDINATEUR
...- --- - .-. . --- .-.
-.. .. -. .- - . ..- .-.
MESSAGE
?
```

Le programme place dans la variable A$ la liste des codes morse de tout l'alphabet. Chaque code doit prendre quatre caractères ; des blancs sont ajoutés pour les codes ne comprenant qu'un, deux ou trois points ou tirets.

```
10 REM MORSE
20 REM ------------------------------
30 LET A$=".-  -...-.-.-.. .   ..-.--. "
40 LET A$=A$+"......  .----.- .-..--  "
50 LET A$=A$+"-.  --- .--.--.-.-. ... -   "
60 LET A$=A$+"..- ...-.-- -..--.----.."
```

Puis la ligne 130 saisit au clavier l'option désirée.

```
70 REM -----------------
80 REM MENU
90 PRINT
100 PRINT "1 ALPHABET"
110 PRINT "2 MESSAGE"
120 PRINT "VOTRE CHOIX "
130 INPUT W
140 IF W=1 THEN 330
150 IF W=2 THEN 380
160 GOTO 120
```

Le sous-programme des lignes 190 à 300 édite le message contenu dans la variable M$.

Chaque caractère est pris séparément (boucle I) et retranscrit en morse (boucle K) en prenant dans la variable A$ la correspondance des lettres.

```
170 REM --------------------
180 REM EDITION MESSAGE
190 FOR I=1 TO LEN(M$)
200 LET P=ASC(MID$(M$,I,1))
210 IF P<65 OR P>90 THEN 280
220 LET P=P-65
230 FOR K=1+P*4 TO 4+P*4
240 IF MID$(A$,K,1)=" "
    THEN 260
250 PRINT MID$(A$,K,1);
260 NEXT K
270 PRINT " ";
280 NEXT I
290 PRINT
300 RETURN
```

Les lignes 330 à 350 placent dans M$ l'alphabet que l'on fait interpréter par le sous-programme des lignes 190 à 300.

```
310 REM ------------
320 REM ALPHABET
330 LET M$="ABCDEFGHI
    JKLMNOPQRSTUVWXYZ"
340 GOSUB 190
350 GOTO 90
```

Les lignes 380 à 420 demandent à l'utilisateur de taper au clavier un message. Il est interprété comme l'alphabet par le sous-programme en 190.

```
360 REM -----------
370 REM MESSAGE
380 PRINT"MESSAGE"
390 INPUT M$
400 IF M$="" THEN 90
410 GOSUB 190
420 GOTO 380
```

Liste des variables

A$	liste des codes morse
I	indice de boucle FOR/NEXT
K	indice de boucle FOR/NEXT
M$	message à émettre
P	pointeur de liste des codes morse
W	choix de l'utilisateur

- LE MORSE - LE MORSE - LE MORSE -

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités: nous vous demandons de consulter votre manuel avant de nous les signaler.

```
10 REM MORSE
20 REM ------------------------------
30 LET A$=".-  -...-.-.-.. .   ..-.--. "
40 LET A$=A$+"......  .----.- .-..--  "
50 LET A$=A$+"-.  --- .--.--.-.-. ... -   "
60 LET A$=A$+"..- ...-.-- -..--.----.."
70 REM ------------------------------
80 REM MENU
90 PRINT
100 PRINT "1 ALPHABET"
110 PRINT "2 MESSAGE"
120 PRINT "VOTRE CHOIX "
130 INPUT W
140 IF W=1 THEN 330
150 IF W=2 THEN 380
160 GOTO 120
170 REM ---------------------------
180 REM EDITION MESSAGE
190 FOR I=1 TO LEN(M$)
200 LET P=ASC(MID$(M$,I,1))
210 IF P<65 OR P>90 THEN 280
220 LET P=P-65
230 FOR K=1+P*4 TO 4+P*4
240 IF MID$(A$,K,1)=" " THEN 260
250 PRINT MID$(A$,K,1);
260 NEXT K
270 PRINT " ";
280 NEXT I
290 PRINT
300 RETURN
310 REM ---------------------------
320 REM ALPHABET
330 LET M$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
340 GOSUB 190
350 GOTO 90
360 REM ---------------------------
370 REM MESSAGE
380 PRINT"MESSAGE"
390 INPUT M$
400 IF M$="" THEN 90
410 GOSUB 190
420 GOTO 380
```

LA BOÎTE AUX LETTRES

Sylvie, un Joseph, un Antoine et un Pierre-Louis ont téléphoné pour t'emmener au ciné-club ce soir. Patou, passe chez le boulanger : deux baguettes... Morceaux de papier épinglés qui s'envolent ou se déchirent, ardoise qui s'efface ? Fini ! Voici la messagerie familiale, boîte aux lettres sur ordinateur.

par Jacques Deconchat

Tel quel, ce programme vous permettra d'enregistrer et de restituer une trentaine de messages de longueur quelconque, en désignant pour chacun d'entre eux un destinataire (il est prévu ici dix destinataires différents, les messages adressés à la collectivité ayant comme destinataire : « Tous »). Le programme est d'une grande facilité d'utilisation, un menu vous désignant les cinq options possibles. Et après chaque utilisation, il revient systématiquement au menu. Si, par accident, on quitte le programme et que l'on veuille y revenir sans effacer les messages en cours, il conviendra de taper :
GOTO 100

```
10 REM MESSAGERIE FAMILIALE
20 DIM M$(30)
30 DIM N$(10)
40 LET N=0
```

Les quatre lignes permettent de réserver la place pour trente messages (variable indicée M $) pour dix noms (variable N $) et de mettre à 0, en début de programme, le compteur de messages N.

```
100 CLS
110 PRINT
120 PRINT "         MESSAGERIE"
130 PRINT
140 PRINT
150 PRINT "  VOUS POUVEZ :"
160 PRINT
170 PRINT "1 LIRE TOUS MESSAGES"
180 PRINT "2 LIRE VOS MESSAGES"
190 PRINT "3 LAISSER UN MESSAGE"
200 PRINT "4 ANNULER UN MESSAGE"
210 PRINT "5 FIN"
220 PRINT
230 PRINT "      VOTRE CHOIX"
240 INPUT R
250 IF R<1 OR R>5 THEN GO
    600.700
260 ON R GO TO 300,400,500,
    TO 100
```

C'est la partie principale du programme, appelée menu. Elle propose cinq options (il sera facile d'en ajouter d'autres), chacune de ces options pouvant être obtenue par un numéro de 1 à 5. En début d'utilisation, c'est évidemment l'option 3 qu'il faudra choisir pour entrer vos messages.

```
500 REM LAISSER UN MESSAGE
510 CLS
520 PRINT "QUEL EST LE
    DESTINATAIRE"
530 INPUT N$(N+1)
550 LET N=N+1
560 PRINT "QUEL EST LE MESSAGE"
570 INPUT M$(N)
580 GO TO 100
```

L'option 3 envoie à la ligne 500, l'écran est effacé (ici CLS, en 510) et le nom du destinataire est demandé en ligne 520. Il faut répondre « TOUS » si le message s'adresse à tous les membres de la collectivité. Le compteur de messages N est ensuite incrémenté d'une unité, le message est tapé, et on retourne au menu.

```
300 REM LECTURE DE TOUS
    LES MESSAGES
310 CLS
320 FOR I=1 TO N
330 PRINT I;"  :";N$(I);
    ":";M$(I)
340 NEXT I
350 INPUT R$
360 GO TO 100
```

Cette option permet de lire la totalité des messages disponibles, il s'agit d'une simple boucle d'affichage. L'arrêt en fin d'affichage est obtenu par un INPUT (ligne 350) : on devra taper un caractère quelconque pour revenir au menu.

```
400 REM LECTURE DE VOS
    MESSAGES
405 CLS
410 PRINT "QUEL EST VOTRE NOM"
420 INPUT R$
430 FOR I=1 TO N
440 IF R$=N$(I) OR N$(I)=
    "TOUS" THEN PRINT I;"
    :";N   $(I);":";M$(I)
450 NEXT I
460 INPUT R$
470 GO TO 100
```

C'est la boucle de lecture sélective des messages. Vous devrez donner votre nom. Les messages qui vous sont précisément adressés, ainsi d'ailleurs que les messages collectifs (« TOUS »), seront alors affichés. La sélection est assurée par la ligne 440.
On pourra n'obtenir que les messages collectifs en tapant un nom inconnu ou « TOUS ».

```
600 REM SUPPRIMER UN MESSAGE
610 CLS
620 PRINT "NUMERO A SUPPRIMER"
630 INPUT L
640 FOR I=L TO N-1
650 LET N$(I)=N$(I+1)
660 LET M$(I)=M$(I+1)
670 NEXT I
680 LET N=N-1
690 GO TO 100
700 REM FIN
```

La suppression d'un message sera faite en donnant le numéro du message à supprimer. Le programme réalise alors un décalage de tous les numéros (si, par exemple, on veut supprimer le message n° 5, l'ancien n° 6 portera le n° 5, le n° 7 deviendra le n° 6, etc.).
La ligne 700 indique la fin du programme. Elle correspond aussi à la fin d'utilisation, obtenue par l'option 5.

LA BOITE AUX LETTRES - LA BOITE AU

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler

```
 10 REM MESSAGERIE FAMILIALE
 20 DIM M$(30)
 30 DIM N$(10)
 40 LET N=0
100 CLS
110 PRINT
120 PRINT "          MESSAGERIE"
130 PRINT
140 PRINT
150 PRINT "  VOUS POUVEZ :"
160 PRINT
170 PRINT "1 LIRE TOUS LES MESSAGES"
180 PRINT "2 LIRE VOS MESSAGES"
190 PRINT "3 LAISSER UN MESSAGE"
200 PRINT "4 ANNULER UN MESSAGE"
210 PRINT "5 FIN"
220 PRINT
230 PRINT "      VOTRE CHOIX"
240 INPUT R
250 IF R<1 OR R>5 THEN GO TO 100
260 ON R GO TO 300,400,500,600,700
300 REM LECTURE DE TOUS LES MESSAGES
310 CLS
320 FOR I=1 TO N
330 PRINT I;"  :";N$(I);":";M$(I)
340 NEXT I
350 INPUT R$
360 GO TO 100
400 REM LECTURE DE VOS MESSAGES
405 CLS
410 PRINT "QUEL EST VOTRE NOM"
420 INPUT R$
430 FOR I=1 TO N
440 IF R$=N$(I) OR N$(I)="TOUS" THEN PRINT I;"  :";N
$(I);":";M$(I)
450 NEXT I
460 INPUT R$
470 GO TO 100
500 REM LAISSER UN MESSAGE
510 CLS
520 PRINT "QUEL EST LE DESTINATAIRE"
530 INPUT N$(N+1)
550 LET N=N+1
560 PRINT "QUEL EST LE MESSAGE"
570 INPUT M$(N)
580 GO TO 100
600 REM SUPPRIMER UN MESSAGE
610 CLS
620 PRINT "NUMERO A SUPPRIMER"
630 INPUT L
640 FOR I=L TO N-1
650 LET N$(I)=N$(I+1)
660 LET M$(I)=M$(I+1)
670 NEXT I
680 LET N=N-1
690 GO TO 100
700 REM FIN
```



LE FLAMBEUR

Las Basicas, 1 heure du matin. La fumée vous empêche de distinguer clairement l'écran de votre partenaire. Un léger clignotement sur les diodes de contrôle laisse deviner sa nervosité. La tension monte... Soudain, le drame. Toutes les lumières s'éteignent. La panne. Cette fois encore, vous vous en tirez sans trop de casse. Mais il lui faudra sa revanche, et alors...

par Jacques Deconchat

Non, votre ordinateur ne deviendra pas un brillant joueur de poker avec ce mini-programme. Il s'agit seulement de s'initier à une technique simple permettant de brouiller un jeu de cartes, de tirer des cartes au hasard et de refaire un tirage si nécessaire. Les possibilités de retirage seront d'ailleurs strictement limitées dans cet exemple au remplacement des cinq cartes tirées par cinq autres, choisies dans les cartes restant disponibles. Il sera néanmoins possible de modifier le programme de façon à pouvoir choisir les cartes que l'on désire garder (un repérage par numéro a été prévu à cet effet). Il n'y a pas non plus d'analyse du jeu, et l'ordinateur ne saura pas vous dire s'il est ou non vainqueur.

Ce programme constitue cependant une bonne façon d'aborder les techniques à maîtriser dans ce type de simulation.

```
 10 REM MINI POKER
 20 DIM C(32)
 30 DIM A$(12)
 40 DATA "TREFLE","COEUR" "CARREAU",
        "PIQUE","7","8".  "9","10",
        "VALET","DAME","ROI","AS"
 50 FOR I=1 TO 12
 60 READ A$(I)
 70 NEXT I
 80 FOR I=1 TO 32
 90 LET C(I)=I
100 NEXT I
110 FOR I=1 TO 31
120 LET N=INT (RND*(33-I)+1)
130 LET J=C(N)
140 LET C(N)=C(I)
150 LET C(I)=J
160 NEXT I
```

Comme c'est souvent le cas, la première partie du programme effectue ce que l'on appelle l'initialisation : deux tableaux sont définis. Un tableau numérique C, dimensionné à 32, recevra les cartes, représentées chacune symboliquement par un nombre entier de 1 à 32. Un tableau alphanumérique A$, dimensionné à 12, recevra les noms des cartes et de leurs couleurs, ceci pour permettre un affichage en clair des jeux tirés (ainsi A$(1) = « trèfle », A$(5) = « 7 », etc.).

La boucle I (lignes 80 à 100) réalise une première distribution du jeu, non brouillé. La deuxième boucle I (lignes 110 à 160) effectue le brouillage, par l'échange de deux cartes au hasard, ceci étant réalisé avec l'aide d'une variable auxiliaire J qui permet de mettre de côté la carte à échanger.

```
170>LET A=1
180 GO SUB 300
190 PRINT
200 PRINT
210 PRINT "VOULEZ-VOUS
        UN AUTRE TIRAGE ?"
220 INPUT R$
230 IF R$="OUI" THEN LET A=6
240 GO SUB 300
250 STOP
```

Ces quelques lignes constituent le corps du programme, les affichages étant effectués par appel d'un sous-programme spécifique commençant en ligne 300. Lorsque A = 1, les dix premières cartes sont affichées (soit cinq cartes pour le joueur et cinq pour l'ordinateur) ; si l'on répond « oui » à la question « Voulez-vous un autre tirage ? », la variable A reçoit la valeur 6, ce qui permet de tirer dix nouvelles cartes dans les vingt-deux restantes (si l'on ne répond pas « oui », les mêmes cartes seront réaffichées).

Pour choisir les cartes à garder, il faudrait faire des permutations sur les cartes à garder, avant d'entrer dans le deuxième sous-programme d'affichage.

```
300>CLS
310 PRINT "VOS CARTES    LES MIENNES"
320 PRINT
330 PRINT
340 FOR I=A TO A+4
350 PRINT I-A+1;":";A$(C(I)-8*INT
    ((C(I)/8))+5);A$(INT (C(I)/8)+1);
360 PRINT TAB (16);"-
    ";A$(C(I+10)-8*INT ((C(I+10)/8
    ))+5);A$(INT (C(I+10)/8)+1)
370 NEXT I
380 RETURN
```

Le sous-programme d'affichage constitue la partie la plus délicate de ce programme. La ligne 350 effectue l'affichage en clair des cinq cartes tirées par le joueur, et les cartes de l'ordinateur sont affichées par la ligne 360. Le décodage du nom des cartes et de la couleur est fait au cours de l'affichage, à l'aide de deux formules (il faut extraire le reste de la division par 8 du numéro de la carte pour avoir sa hauteur, 7, 8, 9, 10, valet, dame, roi, as ; la partie entière de la division permettra d'afficher la couleur, trèfle, carreau, cœur ou pique).

Si votre appareil le permet, on pourra définir des sous-programmes pour une représentation plus réaliste des cartes.

```
VOS CARTES                 LES MIENNES
1:AS      PIQUE    - ROI     COEUR
2:VALET   PIQUE    - 10      COEUR
3:ROI     PIQUE    - VALET   CARREAU
4:9       PIQUE    - 7       COEUR
5:8       TREFLE   - 10      TREFLE
```

LE FLAMBEUR - LE FLAMBEUR - LE FLA

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler

```
 10 REM MINI POKER
 20 DIM C(32)
 30 DIM A$(12)
 40 DATA "TREFLE","COEUR","CARREAU","PIQUE",
    "7","8",  "9","10","VALET","DAME",
    "ROI","AS"
 50 FOR I=1 TO 12
 60 READ A$(I)
 70 NEXT I
 80 FOR I=1 TO 32
 90 LET C(I)=I
100 NEXT I
110 FOR I=1 TO 31
120 LET N=INT (RND*(33-I)+1)
130 LET J=C(N)
140 LET C(N)=C(I)
150 LET C(I)=J
160 NEXT I
170 LET A=1
180 GO SUB 300
190 PRINT
200 PRINT
210 PRINT "VOULEZ-VOUS UN AUTRE TIRAGE ?"
220 INPUT R$
230 IF R$="OUI" THEN LET A=6
240 GO SUB 300
250 STOP
300 CLS
310 PRINT "VOS CARTES          LES MIENNES"
320 PRINT
330 PRINT
340 FOR I=A TO A+4
350 PRINT I-A+1;":";A$(C(I)-8*INT
    ((C(I)/8))+5);A$(INT (C(I)/8)+1);
360 PRINT TAB (16);"- ";A$(C(I+10)-8*INT
    ((C(I+10)/8 ))+5);A$(INT (C(I+10)/8)+1)
370 NEXT I
380 RETURN
```



QUI ÊTES-VOUS ?

Pourriez-vous tuer ? Êtes-vous intelligent ? Savez-vous séduire ? Êtes-vous un bon mari ? Aimez-vous Brahms ? La mode est aux tests de tout poil. Revues, journaux, hebdomadaires sont remplis de ces questionnaires plus ou moins sérieux, dont le but est la plupart du temps de distraire ou de faire sourire. Ils peuvent être traités par un ordinateur.

par Jacques Deconchat

La seule ambition du programme présenté ici : offrir un modèle permettant de réaliser très simplement un questionnaire du type test. Il sera très facilement adaptable pour des questionnaires plus complets (ou plus sérieux) sur tout appareil. La seule condition est de disposer d'une instruction de lecture de données (DATA), qui peut éventuellement être simulée (pour le ZX 81 par exemple ; voir les précédentes fiches programmes).

Le mode d'emploi est très simple : une fois le programme lancé, la première question apparaît sur l'écran. Dans telle situation, comment réagissez-vous ? Quatre réponses possibles, chacune étant précédée d'un numéro d'ordre. On devra taper le numéro (1, 2, 3 ou 4) correspondant à la réponse choisie. Une nouvelle question s'affiche alors. A la fin du déroulement du questionnaire (qui ne comprend que quatre questions, pour ne pas trop allonger la liste), un commentaire résumant les choix effectués sera proposé, et le programme s'arrête.

La première partie du programme (l'initialisation) contient sous forme de données (instruction DATA) les questions à poser, chacune suivie des quatre réponses proposées et des points affectés selon la réponse.

```
  10 REM UN TEST
  20 DATA "VOUS ATTENDEZ AU COIN DE LA RUE VOTRE AMI(
E) , QUI DOIT ARRIVER DANS UN QUART D'HEURE . POUR TU
ER LE TEMPS , VOUS ENTREZ :","1 DANS UN BAR","2 DANS
UNE SALLE DE JEU","3 DANS UN MAGASIN DE VETEMENTS","4
 DANS UN BOUTIQUE D'ELECTRONIQUE",1,5,13,9
  30 DATA "DIMANCHE APRES MIDI . VOUS ETES SEUL(E) .
UN(E) AMI(E) VIENT VOUS VOIR . VOUS LUI PROPOSEZ :","
1 UNE PROMENADE A LA CAMPAGNE","2 DE REGARDER L'EMISS
ION SPORTIVE A LA TELE","3 D'ECOUTER UN DISQUE EN PRE
NANT UN VERRE","4 UNE PARTIE DE PING-PONG",14,6,2,10
  40 DATA "IL EST 4 HEURES . LE BUREAU FERME DANS UNE
 DEMI HEURE . TROIS RAPPORTS TRES URGENTS ATTENDENT .
 VOTRE CHEF DE SERVICE ENTRE ET VOUS DONNE UN DOSSIER
 URGENT","1 VOUS COMMENCEZ TOUT DE SUITE","2 VOUS MET
TEZ LE NOUVEAU DOSSIER AU DESSUS DE LA PILE","3 VOUS
ATTENDEZ QU'IL SOIT SORTI POUR RANGER VOS AFFAIRES","
4 VOUS L'EMPORTEZ A LA MAISON",15,7,3,11
  50 DATA "VOTRE ENFANT A DES DIFFICULTES A L'ECOLE .
 IL VOUS DEMANDE DE LUI VENIR EN AIDE","1 VOUS LUI EX
PLIQUER QUE VOUS AVEZ REUSSI TOUT SEUL","2 VOUS LUI F
AITES RECITER SES LECONS","3 VOUS FAITES SON EXERCICE
","4 VOUS LUI CONSEILLER DE FAIRE AUTRE CHOSE",8,16,1
2,4
```

Pour obtenir un programme relativement court, le nombre des réponses envisageables a été fixé à quatre. On pourrait faire varier ce nombre pour chacune des questions, par exemple en introduisant une donnée chiffrée supplémentaire, en début de DATA, qui indiquerait à l'ordinateur le nombre de réponses à proposer. Une boucle de lecture sera alors nécessaire pour afficher les réponses. Dans l'exemple retenu, chaque donnée comporte neuf renseignements : le texte de la question, les quatre réponses proposées, les points attribués pour chacune des réponses. Il sera bien sûr possible de découper les lignes de DATA si votre appareil impose une limitation dans la longueur des lignes ; par exemple en remplaçant 30 DATA .... par 30 DATA .. puis 31 DATA .., 32 DATA ... etc.

```
100 LET S=0
110 DIM A(4)
120 FOR I=1 TO 4
130 READ A$,B$,C$,D$,E$,A(1),A(2),A(3),A(4)
140 PRINT A$
150 PRINT
160 PRINT B$
170 PRINT C$
180 PRINT D$
190 PRINT E$
200 PRINT
210 PRINT "VOTRE CHOIX (1,2,3,4)"
220 INPUT R
230 IF R<1 OR R>4 THEN GO TO 210
240 LET S=S+A(R)
250 CLS
260 NEXT I
```

Le corps principal du programme comporte tout d'abord une mise à zéro du compteur de points S, puis la définition d'une variable indicée A(I), dimensionnée à 4 (ce sera facultatif sur la plupart des machines), qui servira à lire les points affectés à chaque réponse .

Une boucle I (de la ligne 120 à la ligne 260) permettra ensuite la lecture et l'affichage des questions (pour rallonger le questionnaire en ajoutant des lignes de DATA, il conviendra de modifier la valeur finale de l'indice de boucle ; ici 4 puisqu'il n'y a que quatre questions à lire).

La ligne 130 permet de lire les renseignements (au nombre de 9) contenus dans chaque ligne DATA, et les lignes 140 à 210 effectueront les divers affichages nécessaires.

Le choix effectué par le joueur, enregistré dans une variable R en ligne 220, et contrôlé en ligne 230. Le nombre de points correspondant à ce choix sera ajouté dans S par la ligne 240.

Puis l'écran est effacé (ligne 250 : instruction CLS ou autre, selon les appareils) et on pose la question suivante.

```
 270>IF S<15 THEN PRINT "VOUS FERIEZ MIEUX D'ALLER CO
NSULTER UN PSYCHIATRE"
 280 IF S>=15 AND S<22 THEN PRINT "VOUS PRENEZ LA VIE
 DU BON COTE , MAIS VOUS POURRIEZ TOUT DE MEME TRAVAI
LLER UN PEU PLUS"
 290 IF S>=22 AND S<30 THEN PRINT "DOMMAGE QUE VOUS N
'AYEZ PAS UN PEU PLUS D'AMBITION"
 300 IF S>=30 AND S<38 THEN PRINT "VOTRE PHILOSOPHIE
DE LA VIE EST INTERESSANTE . FAITES VOUS AUSSI DU YOG
A?"
 310 IF S>=38 AND S<45 THEN PRINT "VOUS ETES PLUTOT B
IEN DANS VOTRE PEAU . C'EST RARE"
 320 IF S>=45 THEN PRINT "BRAVO . VOUS ETES QUELQU'UN
 D'EXCEPTIONNEL"
```

En fin de questionnaire, une batterie de tests est prévue pour afficher le commentaire final, qui est fonction du nombre de points que vous aurez obtenu.

```
IL EST 4 HEURES . LE BUREAU FERM
E DANS UNE DEMI HEURE . TROIS RA
PPORTS TRES URGENTS ATTENDENT .
VOTRE CHEF DE SERVICE ENTRE ET V
OUS DONNE UN DOSSIER URGENT

1 VOUS COMMENCEZ TOUT DE SUITE
2 VOUS METTEZ LE NOUVEAU DOSSIER
 AU DESSUS DE LA PILE
3 VOUS ATTENDEZ QU'IL SOIT SORTI
 POUR RANGER VOS AFFAIRES
4 VOUS L'EMPORTEZ A LA MAISON

VOTRE CHOIX (1,2,3,4)
```

QUI ETES-VOUS ? - QUI ETES-VOUS ?

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler

```
  10 REM UN TEST
  20 DATA "VOUS ATTENDEZ AU COIN DE LA RUE VOTRE AMI(
E) , QUI DOIT ARRIVER DANS UN QUART D'HEURE . POUR TU
ER LE TEMPS , VOUS ENTREZ :","1 DANS UN BAR","2 DANS
UNE SALLE DE JEU","3 DANS UN MAGASIN DE VETEMENTS","4
 DANS UN BOUTIQUE D'ELECTRONIQUE",1,5,13,9
  30 DATA "DIMANCHE APRES MIDI . VOUS ETES SEUL(E) .
UN(E) AMI(E) VIENT VOUS VOIR . VOUS LUI PROPOSEZ :","
1 UNE PROMENADE A LA CAMPAGNE","2 DE REGARDER L'EMISS
ION SPORTIVE A LA TELE","3 D'ECOUTER UN DISQUE EN PRE
NANT UN VERRE","4 UNE PARTIE DE PING-PONG",14,6,2,10
  40 DATA "IL EST 4 HEURES . LE BUREAU FERME DANS UNE
 DEMI HEURE . TROIS RAPPORTS TRES URGENTS ATTENDENT .
 VOTRE CHEF DE SERVICE ENTRE ET VOUS DONNE UN DOSSIER
 URGENT","1 VOUS COMMENCEZ TOUT DE SUITE","2 VOUS MET
TEZ LE NOUVEAU DOSSIER AU DESSUS DE LA PILE","3 VOUS
ATTENDEZ QU'IL SOIT SORTI POUR RANGER VOS AFFAIRES","
4 VOUS L'EMPORTEZ A LA MAISON",15,7,3,11
  50 DATA "VOTRE ENFANT A DES DIFFICULTES A L'ECOLE
 IL VOUS DEMANDE DE LUI VENIR EN AIDE","1 VOUS LUI EX
PLIQUER QUE VOUS AVEZ REUSSI TOUT SEUL","2 VOUS LUI F
AITES RECITER SES LECONS","3 VOUS FAITES SON EXERCICE
","4 VOUS LUI CONSEILLER DE FAIRE AUTRE CHOSE",8,16,1
2,4
 100 LET S=0
 110 DIM A(4)
 120 FOR I=1 TO 4
 130 READ A$,B$,C$,D$,E$,A(1),A(2),A(3),A(4)
 140 PRINT A$
 150 PRINT
 160 PRINT B$
 170 PRINT C$
 180 PRINT D$
 190 PRINT E$
 200 PRINT
 210 PRINT "VOTRE CHOIX (1,2,3,4)"
 220 INPUT R
 230 IF R<1 OR R>4 THEN GO TO 210
 240 LET S=S+A(R)
 250 CLS
 260 NEXT I
 270 IF S<15 THEN PRINT "VOUS FERIEZ MIEUX D'ALLER CO
NSULTER UN PSYCHIATRE"
 280 IF S>=15 AND S<22 THEN PRINT "VOUS PRENEZ LA VIE
 DU BON COTE , MAIS VOUS POURRIEZ TOUT DE MEME TRAVAI
LLER UN PEU PLUS"
 290 IF S>=22 AND S<30 THEN PRINT "DOMMAGE QUE VOUS N
'AYEZ PAS UN PEU PLUS D'AMBITION"
 300 IF S>=30 AND S<38 THEN PRINT "VOTRE PHILOSOPHIE
DE LA VIE EST INTERESSANTE . FAITES VOUS AUSSI DU YOG
A?"
 310 IF S>=38 AND S<45 THEN PRINT "VOUS ETES PLUTOT B
IEN DANS VOTRE PEAU . C'EST RARE"
 320 IF S>=45 THEN PRINT "BRAVO . VOUS ETES QUELQU'UN
 D'EXCEPTIONNEL"
```



ET VOS CLASSIQUES ?

Mais qu'a donc répondu La Flèche lorsque Harpagon – L'Avare – lui a demandé... ? Sauriez-vous remettre un peu d'ordre dans un texte – connu – un peu chamboulé ? Sans vous précipiter sur vos petits classiques illustrés, bien sûr. Mais en vous aidant de votre bon sens, éventuellement.

par Jacques Deconchat

Ce programme est construit sur une idée tout à fait banale, mais le même principe pourra aisément donner lieu à toute une série de petits exercices, amusants ou utiles. *Votre Ordinateur* vous présente ici un court extrait d'un texte on ne peut plus classique (vous reconnaîtrez facilement *L'Avare*, de Molière). Malheureusement, les phrases proposées ne sont pas dans l'ordre, et le dialogue tel qu'il est affiché n'a pas beaucoup de sens. Il vous est simplement demandé de retrouver l'ordre initial (et logique !) du dialogue. Pour cela, en réponse à la question « Quel est le bon ordre ? », vous devez taper une série de chiffres, en indiquant d'abord le numéro de la phrase que vous croyez être la première, puis le numéro de la deuxième et ainsi de suite. Quand vous avez fini, vous tapez RETURN (ou ENTER, ou ENTREE, selon les appareils) et l'ordre que vous aurez proposé s'affiche. Si ce n'est pas le bon, la même question sera de nouveau posée, et cela jusqu'à ce que vous ayez trouvé l'ordre convenable.
La première partie du programme (de la ligne 10 à la ligne 120) permet simplement de donner les phrases à afficher, dans un ordre quelconque.

```
10 REM CONNAISSEZ VOUS VOS CLASSIQUES ?
15 LET S=0
20 DIM A$(9)
30 LET A$(1)="LA FLECHE :LES AUTRES ?"
40 LET A$(2)="LA FLECHE :HE BIEN ,
   JE SORS"
50 LET A$(3)="HARPAGON :OUI"
60 LET A$(4)="HARPAGON :SORS D'ICI"
70 LET A$(5)="HARPAGON :LES AUTRES"
80 LET A$(6)="LA FLECHE :QUE VOUS
   EMPORTERAIS-JE ?"
90 LET A$(7)="LA FLECHE :LES VOILA"
100 LET A$(8)="HARPAGON :ATTENDS .
   N'EMPORTES TU RIEN ?"
110 LET A$(9)="HARPAGON :VIENS CA , QUE
   JE VOIS . MONTRE MOI TES MAINS"
120 LET B$="428697513"
```

Une table A$ est dimensionnée à 9 en ligne 20 pour contenir les neuf phrases à afficher. Sur le ZX 81 et le ZX Spectrum, il conviendra d'écrire DIM A$(9,60), 60 représentant la plus grande longueur nécessaire pour une phrase. Sur la plupart des appareils, ce dimensionnement serait inutile (avec moins de dix phrases). La variable S, définie en ligne 15, servira à comptabiliser les essais (on pourra d'ailleurs écrire S = 1, au lieu de S = 0). Quand à la variable B$ (ligne 150), il conviendra, si vous voulez jouer honnêtement, de ne pas la regarder de trop près puisque c'est elle qui détermine pour l'ordinateur l'ordre correct.

```
130 FOR I=1 TO 9
140 PRINT I;"  ";A$(I)
150 NEXT I
160 PRINT
170 PRINT "QUEL EST LE BON ORDRE ?"
180 INPUT C$
```

La boucle I (lignes 130 à 150) permet d'afficher les phrases dans le désordre, chaque phrase étant précédée d'un numéro (donné par I) permettant de la désigner aisément.
La question « Quel est le bon ordre ? » est affichée, et la suite des chiffres que vous allez taper sera enregistrée dans une variable C$ (en ligne 180).
La ligne 190 efface ensuite l'écran (l'instruction d'effacement, ici CLS, peut varier d'un appareil à l'autre).

```
190 CLS
200 PRINT "VOICI VOTRE ORDRE"
210 PRINT
220 FOR I=1 TO LEN (C$)
230 PRINT A$(VAL(MID$(C$,I,1)))
240 NEXT I
250 IF C$=B$ THEN GO TO 310
260 LET S=S+1
270 PRINT "CE N'EST PAS LE BON ORDRE"
280 FOR I=1 TO 1000: NEXT I
290 CLS
300 GO TO 130
310 PRINT
320 PRINT "VOUS AVEZ TROUVE L'ORDRE EN
   ";S;" ESSAIS"
```

Une deuxième boucle I (lignes 220 à 240) affiche les phrases dans l'ordre que vous avez proposé. La ligne 270 servira à indiquer que l'ordre choisi n'est pas le bon, si vous vous êtes trompé. Une petite temporisation est prévue pour ce cas en ligne 280 (boucle vide), et l'on recommence alors au début, non sans avoir oublié d'ajouter 1 dans le compteur d'essais (ligne 260). La ligne 320 termine le programme en affichant le nombre d'essais, dans le cas d'une réponse exacte.

ET VOS CLASSIQUES ?-ET VOS CLASS

Ces programmes en Basic sont conçus pour tourner sur presque tous les ordinateurs individuels. Mais votre machine présente des particularités : nous vous demandons de consulter votre manuel avant de nous les signaler

```
 10 REM CONNAISSEZ VOUS VOS CLASSIQUES ?
 15 LET S=0
 20 DIM A$(9)
 30 LET A$(1)="LA FLECHE :LES AUTRES ?"
 40 LET A$(2)="LA FLECHE :HE BIEN , JE SORS"
 50 LET A$(3)="HARPAGON :OUI"
 60 LET A$(4)="HARPAGON :SORS D'ICI"
 70 LET A$(5)="HARPAGON :LES AUTRES"
 80 LET A$(6)="LA FLECHE :QUE VOUS EMPORTERAIS-JE ?"
 90 LET A$(7)="LA FLECHE :LES VOILA"
 100 LET A$(8)="HARPAGON :ATTENDS . N'EMPORTES TU RIE
N ?"
 110 LET A$(9)="HARPAGON :VIENS CA , QUE JE VOIS . MO
NTRE MOI TES MAINS"
 120 LET B$="428697513"
 130 FOR I=1 TO 9
 140 PRINT I;"  ";A$(I)
 150 NEXT I
 160 PRINT
 170 PRINT "QUEL EST LE BON ORDRE ?"
 180 INPUT C$
 190 CLS
 200 PRINT "VOICI VOTRE ORDRE"
 210 PRINT
 220 FOR I=1 TO LEN (C$)
 230 PRINT A$(VAL(MID$(C$,I,1)))
 240 NEXT I
 250 IF C$=B$ THEN GO TO 310
 260 LET S=S+1
 270 PRINT "CE N'EST PAS LE BON ORDRE"
 280 FOR I=1 TO 1000: NEXT I
 290 CLS
 300 GO TO 130
 310 PRINT
 320 PRINT "VOUS AVEZ TROUVE L'ORDRE EN ";S;" ESSAIS"
```

```
1   LA FLECHE  :LES AUTRES ?
2   LA FLECHE  :HE BIEN  , JE SORS
3   HARPAGON  :OUI
4   HARPAGON  :SORS D'ICI
5   HARPAGON  :LES AUTRES
6   LA FLECHE  :QUE VOUS EMPORTERA
IS-JE ?
7   LA FLECHE  :LES VOILA
8   HARPAGON  :ATTENDS  . N'EMPORTE
S TU RIEN ?
9   HARPAGON  :VIENS CA  , QUE JE V
OIS . MONTRE MOI TES MAINS

QUEL EST LE BON ORDRE ?
```

